

# TEMA 1

---

## ***Exigencias Computacionales del Procesamiento Digital de la Información***

---

### **+ CONTEXTO**

*El propósito general de este tema es describir el modelo matemático que subyace a los circuitos combinacionales. Es decir, la representación digital (binaria) de la información y los operadores básicos del álgebra de Boole. Después estudiamos las distintas formas de representar funciones lógicas, los cambios de representación y la minimización.*

*Estas funciones lógicas serán usadas posteriormente para describir el comportamiento externo de todos los circuitos combinacionales, llamados así porque en el caso ideal su respuesta en un instante sólo depende del valor de las entradas en ese mismo instante. Estos circuitos están relacionados con todos aquellos aspectos de la computación encargados de controlar la ruta de*

*datos y de realizar operaciones aritméticas o lógicas con esos datos.*

*Este tema es el primero del texto y su contenido es básico para todos los demás. Sus antecedentes son de naturaleza lógica y los estudia el alumno, en general, de forma concurrente.*

## **++ CONOCIMIENTO PREVIO NECESARIO**

*El nivel de descripción del tema es muy elemental por lo que no se necesitan conocimientos previos específicos. Vamos a introducir la distinción entre procesado analógico y digital y después iremos introduciendo y explicando todos los conceptos a medida que sean necesarios para el desarrollo del tema. Conviene recordar aquí la equivalencia del álgebra de Boole con los temas análogos que, posiblemente, haya estudiado el lector en los textos de Lógica.*

## **+++ OBJETIVOS DEL TEMA**

*Los objetivos del tema deben coincidir con lo que queremos que el alumno sepa al final del tema y esto se puede especificar en extenso, dando la descripción general de cada uno de los objetivos. Tras el estudio de este tema el alumno deberá haber alcanzado los siguientes objetivos:*

**Objetivo 1:** *Distinguir claramente entre el procesado analógico y digital, entendiendo que son dos formas diferentes de representar la información (los datos) y de operar con ellos para obtener otros datos.*

**Objetivo 2:** *Conocer los postulados y teoremas básicos del Álgebra de Boole y saber demostrar los teoremas.*

**Objetivo 3:** *Saber representar funciones lógicas usando distintos tipos de operadores (AND, OR, NOT; sólo NAND; sólo NOR) y saber pasar de una representación a otra. Por ejemplo, de (AND, OR, NOT) a sólo NAND o de NAND a NOR o de términos mínimos (suma de productos) a términos máximos (productos de sumas).*

**Objetivo 4:** *Saber analizar un circuito lógico. Es decir, saber pasar del esquema de un circuito a la expresión o expresiones lógicas que enlazan las variables de entrada al circuito con las variables de salida.*

**Objetivo 5:** *Saber sintetizar un circuito lógico. Es decir: (a) saber pasar de un conjunto de especificaciones funcionales a una tabla de verdad, (b) de una tabla de verdad a una o más funciones lógicas y (c) de estas funciones al circuito que las satisface.*

**Objetivo 6:** *Saber minimizar funciones lógicas. Es decir, dada una cierta función lógica, encontrar otra equivalente (con la misma tabla de verdad) pero con menos términos o con términos con menos variables. Este objetivo está enlazado con el objetivo 2, porque el proceso de minimización se basa en el uso adecuado y repetido de los postulados y teoremas del Álgebra de Boole. Sin embargo, para alcanzarlo es conveniente usar el conocimiento adicional del método de Karnaugh que recoloca los distintos términos de forma tal que hace evidente el proceso de minimización para funciones de hasta 4 ó 5 variables.*

## ++++ GUÍA DE ESTUDIO

*Al ser el primer tema, las recomendaciones referentes al contenido son las de identificar en cada momento en qué objetivo nos encontramos y usar ejemplos para comprobar nuestras propuestas de minimización, análisis y síntesis. Siempre usamos ejemplos con pocas variables porque así es relativamente sencillo comprobar las funciones que calculan los distintos circuitos, los postulados, los teoremas y los resultados de una minimización usando las tablas de verdad y/o los diagramas de Venn. Es decir, comprobando para todas y cada una de las configuraciones de entrada, cuál es la configuración de salida (inducción completa).*



---

## ***Contenido del Tema***

---

### **1.1. PROCESAMIENTO DIGITAL DE LA INFORMACIÓN**

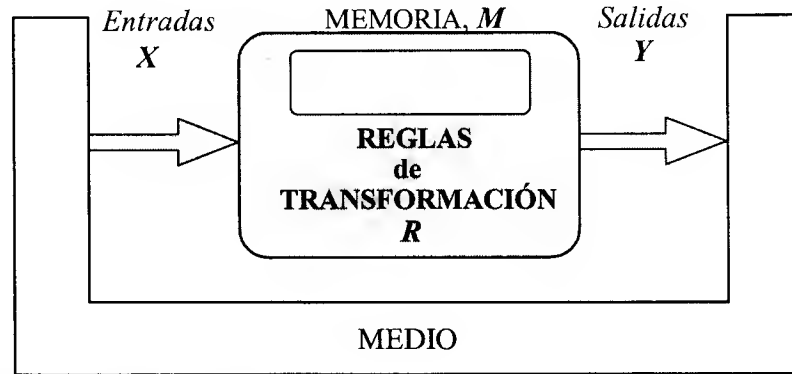
Todo sistema que procesa información puede ser descrito de acuerdo con el modelo computacional de la *figura 1.1*. Se comienza distinguiendo entre *medio* y *sistema* y el comportamiento del sistema se describe en términos de un conjunto de *variables de entrada*,  $X = \{x_i(t)\}$ , un conjunto de *variables de salida*,  $Y = \{y_j(t)\}$ , y un conjunto de *reglas de transformación*,  $R = \{r_{ij}^k(t)\}$ , que realizan procesos sobre las variables de entrada y los *contenidos de memoria*,  $M = \{m_k(t)\}$ , para generar los valores de las variables de salida y modificar los contenidos de memoria. Toda computación puede describirse entonces en términos de un conjunto de *señales*,  $(X, Y, M)$ , que representan la información y de un conjunto de reglas,  $R$ , que especifican los procesos que se usan para transformar la representación de entrada,  $X$ , en la correspondiente representación de salida,  $Y$ .

Las variables de *entrada* y *salida* son magnitudes físicas, en general señales eléctricas (tensiones o corrientes), que sirven de soporte material de la información que se va a procesar. Estas señales pueden ser analógicas o digitales. Una señal analógica,  $x_A(t)$ , admite valores de un espectro continuo entre ciertos valores extremos que marcan su rango dinámico. Por ejemplo, puede ser una señal periódica de tipo sinusoidal que varía entre +5voltios y -5voltios,  $x_A^I(t) = 5 \cos \omega t$ , o una señal en forma de rampa,  $x_A^2(t) = 2t$ .

Por el contrario, una señal digital,  $x_D(t)$ , sólo tiene dos valores posibles, "alto" o "bajo", asociados a dos estados lógicos  $x_D = "1"$ ,  $x_D = "0"$ . Cualquier magnitud analógica admite una representación digital equivalente y viceversa. Es importante señalar que el valor físico real de los dos estados lógicos no es relevante a nivel conceptual. Por ejemplo, el "1" lógico puede estar asociado a  $x_D = 5 \text{ V}$  y el "0" lógico a  $x_D = -5 \text{ V}$ . O bien, el "1" a  $x_D = 1 \text{ V}$  y el "0" a  $x_D = 0 \text{ V}$ . Veremos más adelante



cómo la selección de unos u otros valores está asociada a ciertos criterios de naturaleza electrónica, tales como la velocidad de conmutación, el tipo de transistor usado en la síntesis de los operadores lógicos (bipolar o MOS), la familia lógica (TTL o ECL, por ejemplo) o la inmunidad al ruido.



**Figura 1.1.** Modelo computacional básico.

De la misma forma en que la información puede representarse mediante señales analógicas (continuas) o digitales (binarias), las reglas computacionales,  $\{R_A, R_D\}$ , también pueden ser analógicas o digitales. Una *regla de computación analógica*,  $R_A$ , produce señales de salida,  $y(t)$ , continuas combinando las entradas y los contenidos de memoria mediante operaciones analógicas lineales o no lineales. Por ejemplo, sumando, multiplicando por una constante, derivando, integrando, etc...). Así, las siguientes expresiones son ejemplos de computación analógica:

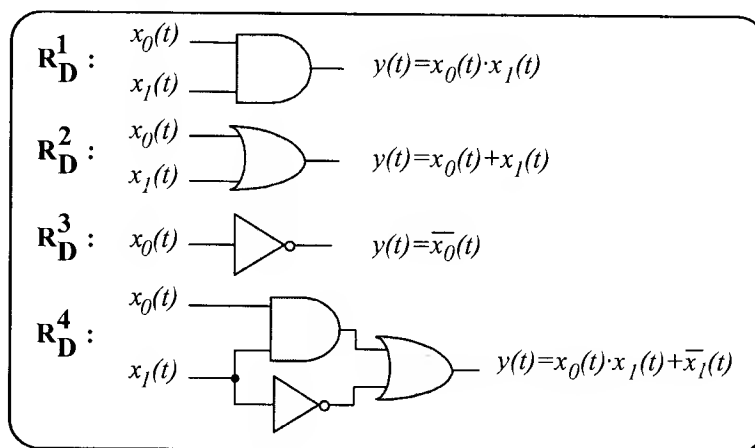
$$R_A^1 : y(t) = A \cdot x(t)$$

$$R_A^2 : y(t) = A_1 \cdot x_1(t) + A_2 \cdot x_2(t)$$

$$R_A^3 : y(t) = A_1 \cdot x_1(t) + B \int x_2(t) \cdot dt + C \frac{dx_3(t)}{dt}$$

De forma equivalente, una *regla de computación digital*,  $R_D$ , produce representaciones binarias de salida (vectores lógicos de "ceros" y "unos"), combinando la representación binaria de entrada,  $X_D(t)$ , con el contenido de memoria,  $M_D(t)$ , mediante operadores digitales. Un conjunto completo de estos operadores lo forman la suma lógica (operador "OR"), el producto lógico (operador "AND") y la

inversión o complemento de una variable (operador NOT). Así, las siguientes expresiones (*figura 1.2*) son ejemplos de computación digital, que se muestran asociadas al símbolo que las representa.



**Figura 1.2.** Expresiones de computación digital.

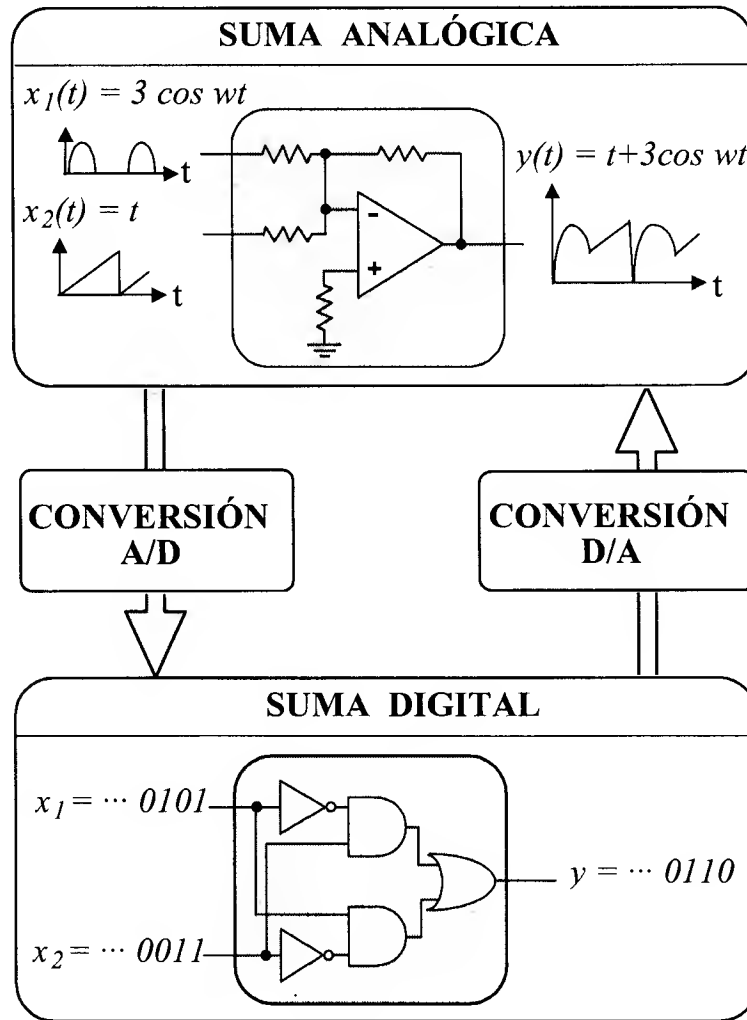
Como las magnitudes binarias sólo pueden poseer uno de dos valores, (0,1), los operadores digitales se pueden representar mediante "*tablas de verdad*" que especifican de forma completa, en extenso, el valor que toma la salida para cada una de las posibles configuraciones de valores en la entrada. Así, para dos variables,  $x_0(t)$ ,  $x_1(t)$ , hay cuatro posibilidades de composición:  $(x_0=0, x_1=0)$ ,  $(x_0=1, x_1=0)$ ,  $(x_0=0, x_1=1)$  y  $(x_0=1, x_1=1)$ , con lo cual la especificación de las reglas usadas como ejemplo es la que se muestra en la *figura 1.3*.

$x_1$	$x_0$	$\overline{x_1}$	$x_1 \cdot x_0$	$x_1 + x_0$	$x_1 \cdot x_0 + \overline{x_1}$
0	0	1	0	0	1
0	1	1	0	1	1
1	0	0	0	1	0
1	1	0	1	1	1

**Figura 1.3.** Tablas de verdad que especifican en extenso los operadores lógicos usados en el ejemplo a partir de las cuatro configuraciones de valores posibles en las dos variables de entrada,  $(x_0, x_1)$ .

La *figura 1.4* ilustra la representación y el procesado digital de la información comparando con

la solución analógica equivalente para el caso de una sencilla suma. Se incluye también las transformaciones que serían necesarias para cambiar de representación, es decir, las conversiones analógicas-digitales y digitales-analógicas.



**Figura 1.4.** Ilustración cualitativa de la suma analógica comparada con la suma digital

En este texto nos vamos a preocupar de hacer un análisis del conjunto mínimo de operadores digitales necesarios para la síntesis de cualquier procesador digital y de estudiar después su síntesis modular a nivel lógico para incluir finalmente la realización física de esos módulos básicos en las

tecnologías bipolar, MOS y C-MOS. Afortunadamente, el diseño digital y por consiguiente la electrónica que lo soporta son *modulares*, de forma que un conjunto completo de operadores lógicos sólo necesita tres circuitos: AND, OR y NOT. Veremos más adelante que basta una combinación de dos de estos tres operadores (NAND ó NOR) para poder sintetizar cualquier función lógica.

## 1.2. FUNCIONES COMBINACIONALES Y SECUENCIALES NECESARIAS.

La electrónica digital es una ingeniería. Es decir, su tarea fundamental es la síntesis de circuitos que satisfacen determinadas especificaciones funcionales. Primero aparece una necesidad de procesamiento digital de la información y después buscamos un circuito que satisface esa necesidad siguiendo, aproximadamente, los siguientes pasos:

1. *Descripción de la computación en lenguaje natural de forma clara, completa, precisa e inequívoca.*
2. *Traslación de esa descripción a un conjunto de especificaciones funcionales en un lenguaje lógico formal.*
3. *Reescritura de esa descripción formal en términos del modelo computacional de la figura 1.1. Es decir, en términos de entradas, estados de memoria, salidas y reglas de transformación,  $\{R_D\}$ , que representan la dinámica interna de la computación especificando cómo se producen el nuevo estado y las salidas a partir de las entradas y del estado anterior.*
4. *Síntesis modular del sistema en términos de un conjunto completo de operadores mínimos.*

Todas las funciones necesarias para el procesamiento digital de la información, es decir todas las reglas  $R_D$ , pueden incluirse en dos grandes apartados:

- a) *Funciones Combinacionales.*
- b) *Funciones Secuenciales.*

Son funciones de *lógica combinatorial* todas aquellas funciones en las que para obtener el valor de la salida en un cierto instante sólo necesitamos conocer el valor de las entradas en ese mismo

instante. Son funciones de decisión, sin "memoria". El modelo matemático soporte de esta parte de la electrónica digital es el *Álgebra de Boole*, de la que más adelante incluiremos un resumen. Ejemplos de este tipo de funciones son entre otras:

- a.1. *Operaciones aritmético-lógicas.*
- a.2. *Funciones de ruta de datos: multiplexos y demultiplexos.*
- a.3. *Circuitos cambiadores de código.*

Es cierto que, aunque en el caso ideal las funciones combinacionales son “instantáneas”, a la hora de sintetizar físicamente estas funciones usando circuitos integrados reales aparecen pequeños retardos no deseados. El objetivo del diseño en electrónica digital combinacional es hacer mínimo ese retardo que limita la velocidad de conmutación.

Aparece así el primer conjunto de necesidades computacionales. Necesitamos saber *representar*, *analizar* y *sintetizar* circuitos lógicos combinacionales. Es importante que el alumno distinga bien estas tres tareas.

**¿Qué es representar?** Representar una función lógica combinacional es encontrar un procedimiento para *describir de forma completa* la función. Sea cual fuere la configuración de valores en las variables de entrada, la representación debe permitir conocer el valor de la salida. Hay esencialmente dos formas de representación: en *extenso* y en *intenso*.

Decimos que una función combinacional está representada en *extenso* cuando poseemos una tabla con todos los valores de las entradas y un vector lógico de salida que nos dice para cada configuración (00··0, 00··1, ..., 11··1), si la salida debe estar en alta (1) ó en baja (0). Las dos formas más usuales de representar funciones lógicas en extenso son las *tablas de verdad* y los *diagramas de Venn*, en los que a cada variable lógica se le asocia un área dentro de un rectángulo.

Decimos que tenemos una función combinacional representada en *intenso* cuando hay una expresión booleana que la describe. Por ejemplo,  $f(x, y, z) = (\overline{x}\overline{y} + xy)\overline{z}$ . Lógicamente, al dar valores a las variables (x, y, z) obtenemos la tabla en extenso.

**¿Qué es analizar?** Analizar un circuito en lógica combinacional es encontrar la representación de las funciones lógicas que lo describen. En el sencillo ejemplo de suma digital mostrado en la *figura 1.4* el proceso de análisis debe darnos la expresión lógica de la relación entre las variables

de entrada,  $(x_1, x_2)$  y la salida,  $y$ . Siguiendo el camino de las señales es fácil ver que en este caso la función  $y = f(x_1, x_2)$  es de la forma:

$$y = f(x_1, x_2) = (x_1 \bar{x}_2 + \bar{x}_1 x_2) \quad [1.1]$$

**¿Qué es sintetizar?.** Es el *proceso inverso al de análisis*. Partimos ahora de una función lógica,  $y=f(x_1, x_2)$ , ó de un conjunto de  $N$  funciones lógicas de  $M$  variables,  $y_k(t)=f_k[x_1(t), x_2(t), \dots, x_M(t)]$ ,  $k=1, 2, \dots, N$ , y buscamos el circuito que realiza físicamente a esa función de forma que reproduce la misma relación entre las variables de entrada y las de salida para todas y cada una de las combinaciones de valores posibles.

El problema de síntesis en sistemas digitales es más complicado que el de análisis porque el punto de partida de la síntesis de un determinado circuito lógico no está especificado con la claridad y precisión de la lógica sino que parte de una descripción en lenguaje natural. Por ejemplo:

*Diseñar un circuito que dé paso a la orden de arranque de un coche sólo cuando las puertas ( $x_1$ ) estén cerradas ( $x_1=1$ ), el cinturón de seguridad ( $x_2$ ) abrochado ( $x_2=1$ ), las luces ( $x_3$ ) apagadas ( $x_3=1$ ) y el motor ( $x_4$ ) no esté ya en marcha ( $x_4=0$ ).*

Es decir, los puntos 1º y 2º del procedimiento general son importantes y nunca son tan sencillos como en el ejemplo del coche. Una vez que tenemos una descripción clara, precisa y completa de una función combinatorial, es relativamente sencillo encontrar el circuito que la realiza físicamente. Así, en el ejemplo del coche la función sería:  $y = x_1 x_2 x_3 \bar{x}_4$ . No sólo aparecen dificultades a la hora de asignar variables lógicas a frases del lenguaje natural (“puerta abierta o cerrada”, “luces encendidas”, “motor en marcha”, ...) sino que también aparecen dificultades en la tarea de síntesis digital a la hora de obtener un conjunto completo de especificaciones funcionales claras, precisas e inequívocas (un “algoritmo”) del método adecuado para resolver una tarea. Por ejemplo:

1. *Diseñar un circuito que realice operaciones aritméticas en coma flotante para palabras de 16 bits.*
2. *Diseñar un multiplicador en complemento a 2 de alta velocidad.*

Estas descripciones iniciales están más próximas a la lógica y a la teoría de autómatas que a las

del ejemplo del coche pero necesitan mucho conocimiento sobre formas de representación de números positivos y negativos, distintos algoritmos de multiplicación, etc... Su estudio, hasta conocer los circuitos intermedios básicos, es objeto de la Electrónica Digital. A partir de un cierto momento, cuando estos circuitos se integran en arquitecturas más complejas, su estudio corresponde a la Arquitectura de Computadores o a las otras grandes áreas de aplicación de la Electrónica Digital: *Comunicación, Control e Instrumentación*.

Hemos visto una primera justificación de las necesidades computacionales en lógica combinacional. Sin embargo, hay otro gran apartado de necesidades de computación en las que es imprescindible introducir el *tiempo* como variable de *cálculo*. A los circuitos correspondientes se les llama *circuitos secuenciales* y su modelo matemático es la *Teoría de Autómatas Finitos*. Representan aquellas situaciones en las que para calcular la respuesta de un sistema en un cierto instante,  $t = t_n$ , no basta con conocer las entradas en ese instante, sino que necesitamos conocer su "*estado*" y por consiguiente sus entradas y salidas en instantes anteriores. Decimos entonces que el sistema tiene "*memoria*".

Ejemplos de funciones secuenciales básicas en la electrónica digital son, entre otros:

**b.1.** *Contadores.*

**b.2.** *Registros de desplazamiento.*

**b.3.** *Temporizadores*

**b.4.** *Memorias RAM.*

Para la síntesis de estas funciones secuenciales necesitamos los mismos operadores que usamos en la lógica combinacional, más la representación computacional del retardo. Es decir, una función  $Q(t_n)$  que coincide con el valor de la entrada,  $D$ , en el instante anterior

$$Q(t) = D(t - \Delta t) \quad \text{o bien} \quad Q(t_n) = D(t_{n-1})^{(*)} \quad [1.2]$$

Esta función la realiza un circuito biestable tipo D (de "*Delay*"). Existen otros tipos de

---

(\*) A lo largo del texto usaremos de forma indistinta una u otra notación. En algunas ocasiones para simplificar también usaremos  $Q_n = D_{n-1}$ . Es decir,  $D(t - \Delta t)$ ,  $D(t_{n-1})$  y  $D_{n-1}$  significan lo mismo: el valor de  $D$  en el instante de tiempo anterior al que se está considerando en este momento.

biestables (T, R-S y J-K) que incluyen también el retardo pero con funciones de disparo diferentes.

Veremos en su momento que estos circuitos biestables se pueden realizar mediante el acoplo de dos inversores con realimentación positiva, haciendo que siga siendo cierto que toda la electrónica digital se puede sintetizar con un único tipo de operador.

Seguimos avanzando en la contestación a la pregunta que daba título a este apartado. Es decir, estamos enumerando el conjunto de funciones necesarias para la representación, análisis y síntesis de sistemas digitales. Una relación de esas necesidades, razonablemente completa, debería incluir al menos a las siguientes:

- a) *Funciones combinacionales*, cuyo modelo matemático es el álgebra de Boole que vamos a estudiar a continuación.
- b) *Funciones secuenciales*, cuyo modelo matemático es la teoría de autómatas que estudiaremos más adelante, para aproximarla al tema en el que se usa.
- c) *Funciones de temporización*, tanto síncronas como asíncronas, necesarias para engarzar en coordinen todas las operaciones combinacionales y secuenciales que constituyen un sistema digital. Ejemplos de funciones de temporización son:
  - c.1. *Monoestables*
  - c.2. *Osciladores astables*.
  - c.3. *Temporizadores programables*.
  - c.4. *Relojes monofásicos y polifásicos*.

Una vez que tenemos el panorama aproximado de las funciones necesarias para la síntesis de sistemas digitales, dedicaremos el resto del tema al estudio de las bases matemáticas y el resto del libro al estudio de esas funciones siguiendo un criterio de complejidad creciente.

### **1.3. VARIABLES Y OPERADORES LÓGICOS: ÁLGEBRA DE BOOLE**

Tal como hemos desarrollado en extenso en el apartado anterior, toda la electrónica digital se basa en el uso de variables binarias y operadores lógicos para representar, transformar, almacenar y recuperar información. El modelo matemático soporte es el álgebra de Boole y la meta final será encontrar realizaciones físicas en distintas tecnologías electrónicas de sus operadores básicos.



El álgebra de Boole fue desarrollada inicialmente para el estudio formal del lenguaje. En su libro "*Las Leyes del Pensamiento*" decía George Boole que todas las operaciones del lenguaje, considerado como soporte del razonamiento, pueden hacerse computacionales mediante un sistema de símbolos compuesto por los siguientes elementos:

- a) *Símbolos literales*,  $(X, Y, Z, \dots)$ , que representan *cosas*, como sujetos de nuestras concepciones.
- b) *Símbolos de operación*, como  $(+, \cdot, \text{complemento})$ , que representan operaciones mentales por medio de las cuales combinamos los conceptos o formamos nuevos conceptos referentes a los mismos elementos.
- c) El *signo de identidad*  $(=)$ .

Estos símbolos están sujetos a leyes bien definidas que C.E. Shannon en 1938 y en su trabajo "*Un Análisis Simbólico de los Circuitos con Redes*", asocia a representaciones físicas en términos de niveles de tensión y circuitos conmutadores que conducen o no conducen. Nace así la *Teoría de la Conmutación* y con ella el puente entre la lógica formal y su realización física mediante circuitos electrónicos combinacionales o secuenciales.

Veamos ahora un resumen del Álgebra de Boole de dos valores de clase  $B = \{0, 1\}$ . Si sobre un conjunto de variables binarias  $(X, Y, Z, \dots)$  definimos las operaciones de suma lógica  $(+)$ , producto lógico  $(\cdot)$  y complementación (*inversión*), el conjunto queda dotado de la estructura del Álgebra de Boole si cumple los siguientes postulados:

**P.1.** *Las operaciones  $(+ \text{ y } \cdot)$  son operaciones cerradas.* El resultado de aplicarlas a cualesquiera de las variables del conjunto, producirá variables del conjunto.

**P.1.a:** Si  $X \in B$  é  $Y \in B$ , entonces  $(X+Y) \in B$

**P.1.b:** Si  $X \in B$  é  $Y \in B$ , entonces  $(X \cdot Y) \in B$

[1.3]

**P.2.** *Existen elementos neutros para ambas operaciones ("0" para la suma y "1" para el producto).*

**P.2.a:**  $X + 0 = X$

**P.2.b:**  $X \cdot 1 = X$

[1.4]

**P.3.** *Ambas operaciones son conmutativas:*

$$\mathbf{P.3.a:} \quad X + Y = Y + X$$

$$\mathbf{P.3.b:} \quad X \cdot Y = Y \cdot X$$

[1.5]

**P.4.** *Ambas operaciones son distributivas, una respecto de la otra:*

$$\mathbf{P.4.a:} \quad X + Y \cdot Z = (X + Y) \cdot (X + Z) \quad (\text{La suma es distributiva respecto al producto})$$

$$\mathbf{P.4.b:} \quad X(Y + Z) = X \cdot Y + X \cdot Z \quad (\text{El producto es distributivo respecto a la suma})$$

[1.6]

**P.5.** *Complementariedad.* Para cada variable  $X$  existe su complementaria  $\overline{X}$  y entre ambas se cumplen las siguientes condiciones:

$$\mathbf{P.5.a:} \quad X + \overline{X} = 1$$

$$\mathbf{P.5.b:} \quad X \cdot \overline{X} = 0$$

[1.7]

---

**Ejercicio:** *Demostrar que “el producto es distributivo respecto de la suma”.*

**Solución:**

Su expresión lógica es:

$$X(Y+Z) = XY+XZ \quad [1.8]$$

Podríamos demostrarlo por inducción completa, poniendo las tablas de verdad de ambas expresiones supuestamente equivalentes y viendo después que los dos circuitos correspondientes también son equivalentes: Las *figuras 1.5* muestra las tablas de verdad y la 1.6 los circuitos correspondientes a las expresiones lógicas de ambos lados del igual.

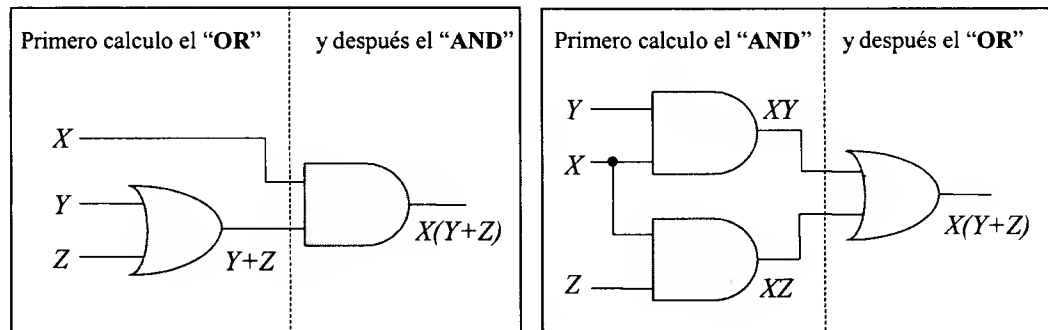
Obsérvese que la columna correspondiente a la parte izquierda del igual,  $X(Y+Z)$ , y la correspondiente a la parte derecha,  $XY+XZ$ , son iguales. También los circuitos nos muestran el

carácter intercambiable del orden de las variables para las operaciones OR y AND.

$X$	$Y$	$Z$	$Y+Z$	$X(Y+Z)$	$XY$	$XZ$	$XY+XZ$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

son iguales

**Figura 1.5.** Tablas de verdad para demostrar la distributividad del producto respecto de la suma.



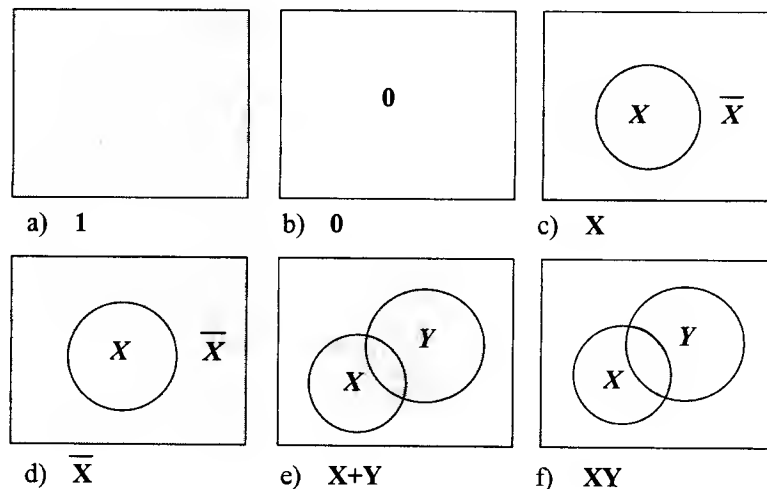
**Figura 1.6.** Circuitos funcionalmente equivalentes.

De forma análoga podríamos demostrar que también la suma es distributiva respecto del producto. Es decir, que  $X+YZ=(X+Y)(X+Z)$ .

Una propiedad importante del álgebra de Boole es la *dualidad*. Cada miembro de un par dual, por ejemplo  $X+0 = X$ , se puede obtener del otro miembro del par,  $(X \cdot 1 = X)$ , intercambiando los elementos "0" y "1" y los operadores suma por producto y viceversa. Si una relación es cierta, también lo será su dual.

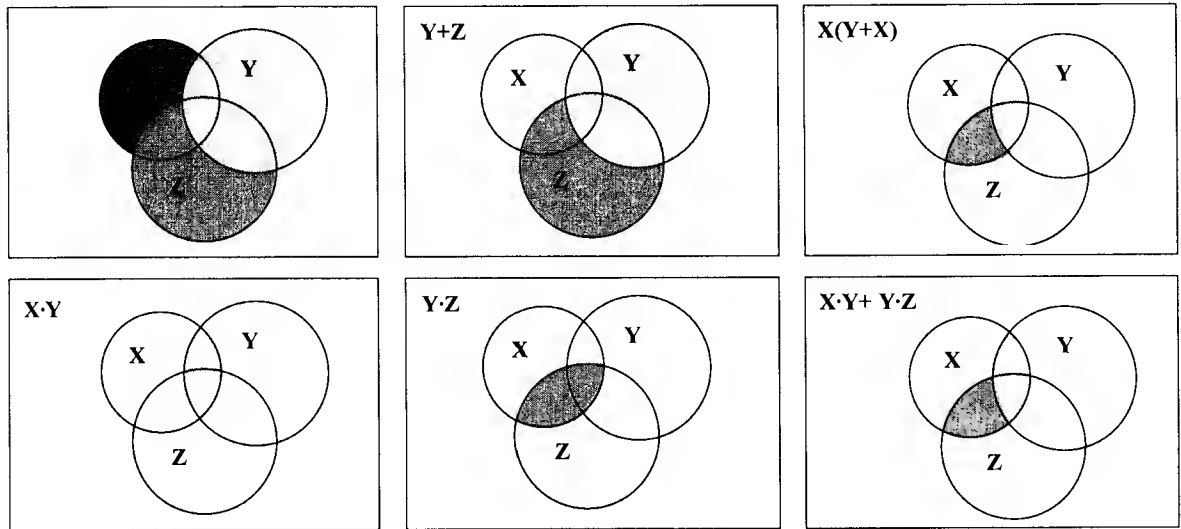
Los operadores suma y producto y la complementación han sido definidos en el apartado anterior mediante sus tablas de verdad. Esta es una definición *en extenso*, es decir describimos la función realizada por un operador listando el conjunto de todas las configuraciones de valores posibles en sus variables de entrada y viendo cual es la respuesta a cada una de esas configuraciones. Esta forma de definir los operadores nos permite usar el procedimiento de *inducción completa* para demostrar los postulados P.2, P.3, P.4 y P.5 y para pronunciarnos acerca de la verdad o falsedad de cualquier otra relación.

La otra forma usual y sencilla de demostrar la verdad o falsedad de una determinada relación lógica para expresiones con pocas variables es usar los diagramas de Venn, basados en la unión e intersección de conjuntos. A cada variable se le asocia un área matizada dentro de un rectángulo. Su complementaria es el resto del rectángulo, de forma que el área del rectángulo representa la unidad. La *figura 1.7* ilustra el procedimiento, para la negación de una variable, ( $\bar{X}$ ), la suma ( $X+Y$ ) que incluye el área que pertenece a  $X$  ó pertenece a  $Y$ , y el producto ( $X \cdot Y$ ) que sólo incluye el área que pertenece a  $X$  y a  $Y$ . Si dos expresiones son iguales deben de generar la misma área matizada.



**Figura 1.7.** Diagramas de Venn

Veamos ahora el mismo ejemplo de *distributividad respecto a la suma* usando los diagramas de Venn. Es decir, veamos que  $X(Y+Z)$  produce la misma área matizada que  $XY+XZ$ . La *figura 1.8* ilustra el proceso:



**Figura 1.8.** Demostración de distributividad del producto respecto de la suma mediante diagramas de Venn.

A partir de los postulados **P.1**, ..., **P.5**, enunciados anteriormente, pueden enunciarse y demostrarse un conjunto de *teoremas*, entre los que podemos destacar los siguientes:

<b>Teoremas</b>	
<b>T.1:</b> Doble complementación	$\overline{\overline{X}} = X$
<b>T.2:</b> Idempotencia	$X + X = X$ $X \cdot X = X$
<b>T.3:</b> Absorción	$X + X \cdot Y = X$ $X \cdot (X + Y) = X$
<b>T.4:</b> Adyacencia	$X \cdot Y + X \cdot \overline{Y} = X$ $(X + Y) \cdot (X + \overline{Y}) = X$
<b>T.5:</b> Teoremas de DeMorgan:	$\overline{X + Y} = \overline{X} \cdot \overline{Y}$ $\overline{X \cdot Y} = \overline{X} + \overline{Y}$

Veamos ahora, recordando los cinco postulados del Álgebra de Boole, cómo podemos

demostrar algunos de estos teoremas.

**Ejercicio:** Demostrar el teorema de “absorción”.

**Solución:**

Las expresiones lógicas del *teorema de absorción* son:

$$X + XY = X \quad \text{y} \quad X(X + Y) = X \quad [1.9]$$

Podemos demostrar este teorema mediante los tres procedimientos vistos anteriormente. Es decir, por tablas de verdad, por aplicación directa de los postulados y mediante diagramas de Venn.

a) Por *tablas de verdad*

$X$	$Y$	$Y \cdot Y$	$X + XY$	$X + Y$	$X(X + Y)$
0	0	0	0	0	0
0	1	0	0	1	0
1	0	0	1	1	1
1	1	1	1	1	1

coinciden

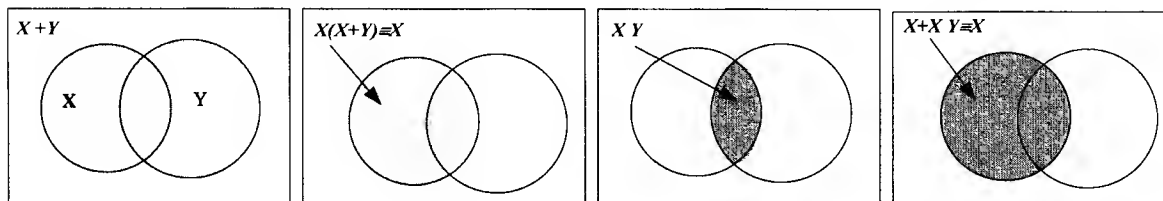
**Figura 1.9.** Demostración del teorema de absorción mediante tablas de verdad.

Obsérvese como coinciden las columnas correspondientes a  $X$  con las correspondientes a  $X + XY$  y  $X(X + Y)$ .

b) Por *aplicación directa de los postulados*

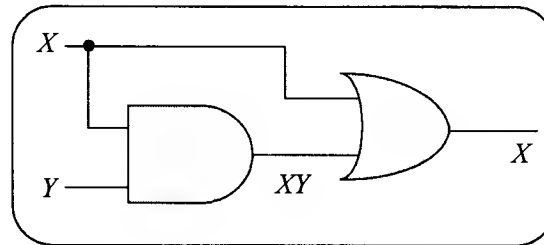
Sacando a la variable  $X$  como factor común. Así,  $X + XY = X(1 + Y) = X$ , puesto que  $1 + Y = 1$ .

c) Por *diagramas de Venn*:



**Figura 1.10.** Demostración del teorema de absorción mediante diagramas de Venn.

Si pensamos en términos de circuitos, la demostración también es evidente, puesto que a la puerta “OR” de salida llegan  $X$  y el producto  $XY$ , que siempre es más restrictivo que sólo  $X$ .



**Figura 1.11.** Demostración mediante el circuito que implementa la expresión lógica del teorema de absorción.

Dejamos para el alumno la demostración de la expresión dual  $X(X+Y)=X$ .

**Ejercicio:** Demostrar el teorema de adyacencia

**Solución:**

El teorema de *adyacencia* se pone de manifiesto mediante las siguientes expresiones:

$$XY + X\bar{Y} = X \quad \text{y} \quad (X+Y) \cdot (X+\bar{Y}) = X \quad [1.10]$$

Ya deberíamos saber actuar sin mucho pensamiento: ponemos las tablas de verdad o dibujamos los diagramas de Venn o aplicamos directamente los postulados y después podemos implementarlo dibujando los circuitos.

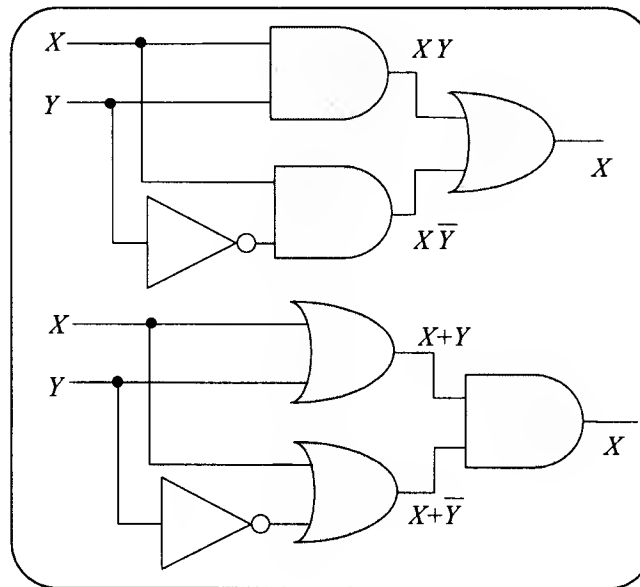
Veámoslo mediante las tablas de verdad.

$X$	$Y$	$XY$	$X\bar{Y}$	$XY+X\bar{Y}$	$X+Y$	$X+\bar{Y}$	$(X+Y) \cdot (X+\bar{Y})$
0	0	0	0	0	0	1	0
0	1	0	0	0	1	0	0
1	0	0	1	1	1	1	1
1	1	1	0	1	1	1	1

*son iguales*

**Figura 1.12.** Demostración de los teoremas de adyacencia.

Como podemos observar las tres columnas coinciden. Veamos ahora los circuitos que implementan ambas funciones.



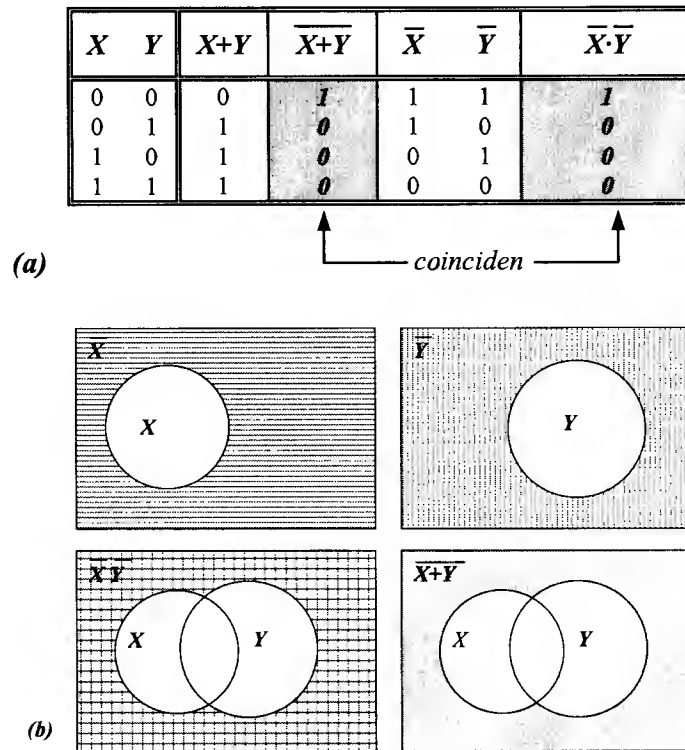
**Figura 1.13.** Implementación de los teoremas de adyacencia.

Obsérvese que el paso de una expresión a su *dual* sólo ha permutado la posición de los circuitos AND con la de los OR.

El primero de los *teoremas de DeMorgan* lo vamos a demostrar por *inducción completa* y el segundo gráficamente, mediante el uso de diagramas de Venn. Los resultados se muestran en la *figura 1.14*. En *1.14.a*, hemos escrito las cuatro configuraciones lógicas asociadas a las dos variables  $X$  e  $Y$  (00, 01, 10, 11) y los estados lógicos correspondientes a  $(X + Y)$ ,  $(\overline{X + Y})$ ,  $\overline{X}$ ,  $\overline{Y}$  y  $\overline{X} \cdot \overline{Y}$ . Obsérvese que la columna correspondiente a  $\overline{X + Y}$  coincide con la correspondiente a  $\overline{X} \cdot \overline{Y}$ , con lo que queda demostrada la igualdad de ambas expresiones:  $\overline{X + Y} = \overline{X} \cdot \overline{Y}$ .

Por otro lado, en la *figura 1.14.b* se parte de las áreas asociadas a  $X$  e  $Y$  y se construyen las áreas asociadas a  $\overline{X}$ ,  $\overline{Y}$ ,  $X \cdot Y$  y  $\overline{X + Y}$ . Obsérvese como el área rayada asociada a  $\overline{X + Y}$  es la misma que la que corresponde a  $\overline{X} \cdot \overline{Y}$ .





**Figura 1.14.** Demostración de los Teoremas de Morgan. (a) Por inducción completa. (b) Por diagramas de Venn.

**Ejercicio:** Usar un simulador para comprobar que las respuestas correspondientes a los circuitos de ambos lados del igual,  $\overline{(X+Y)}$  y  $(\bar{X} \cdot \bar{Y})$ , coinciden

**Solución:**

En la simulación, para generar de forma automática la tabla de verdad del circuito, usamos como señales de entrada dos relojes cuyas frecuencias son una el doble de la otra, así tenemos las cuatro configuraciones posibles para las dos variables  $X$  y  $Y$ . Hemos tomado para  $X$  una frecuencia de 1MHz y para  $Y$  una frecuencia de 2MHz. La figura 1.15 muestra las tablas de verdad, el circuito y el cronograma correspondiente a  $\overline{X+Y} = \bar{X} \cdot \bar{Y}$ .

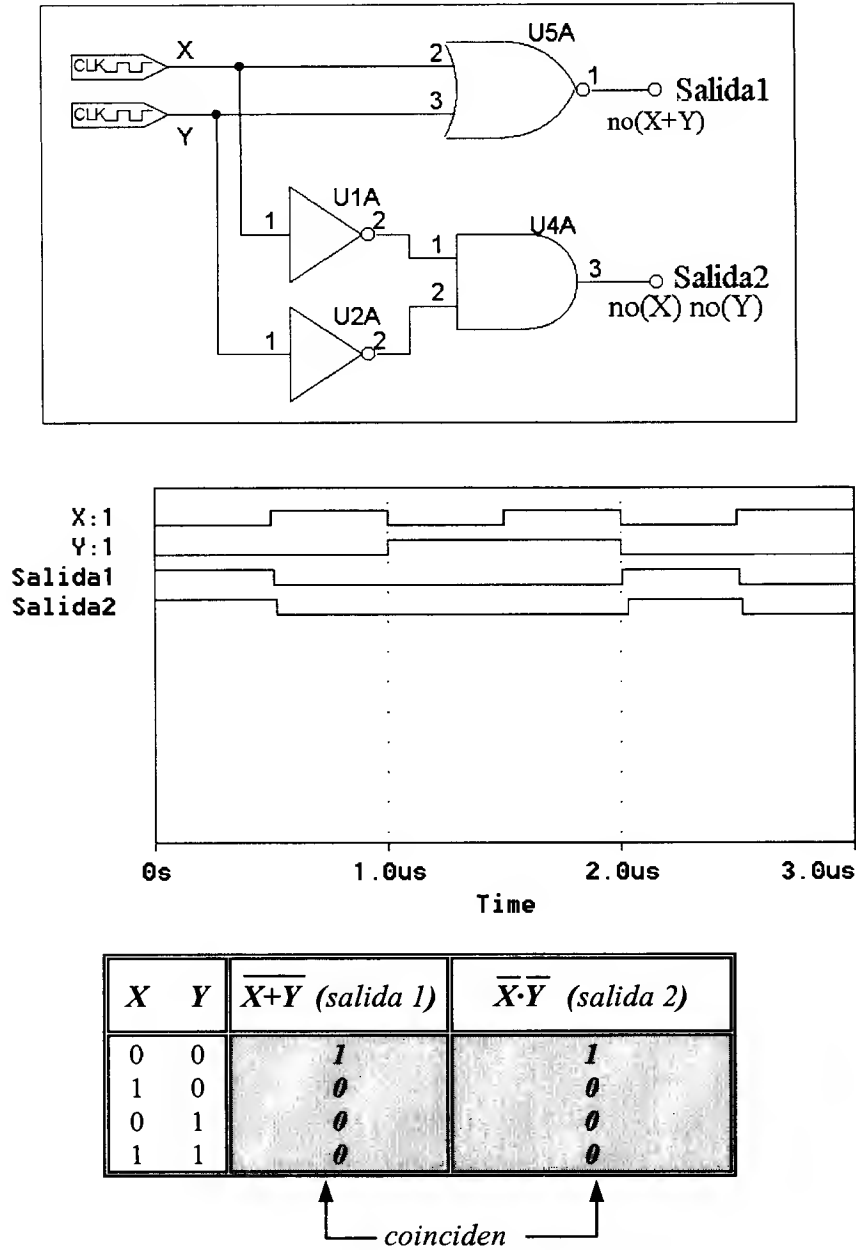


Figura 1.15. Simulación de los circuitos  $\overline{X+Y}$  y  $\overline{X} \cdot \overline{Y}$  (leyendo en vertical los valores de las variables del cronograma , comprobamos la tabla de verdad)

## 1.4. FUNCIONES LÓGICAS: FORMAS CANÓNICAS

Un conjunto de *variables lógicas* conectadas por cualquier combinación de los *operadores básicos* (suma, producto y complementación) constituye una función lógica. Más formalmente, una función lógica,  $f$ , de  $n$  variables,  $(x_1, x_2, \dots, x_n)$ , sobre un álgebra de Boole  $B = \{0,1\}$ , es cualquier aplicación del producto cartesiano de  $B$  por sí mismo  $n$  veces,  $B^n$ , en  $B$ .

$$\begin{array}{c} \xleftrightarrow{n} \\ B^n = B \times B \times \dots \times B \\ B^n \xrightarrow{f(x_1, \dots, x_n)} B \end{array} \quad [1.11]$$

Los elementos de  $B^n$  son configuraciones lógicas de  $n$  variables. Para  $n=2$ , tenemos cuatro configuraciones ( $x_1 = 0, x_2 = 0$ ;  $x_1 = 0, x_2 = 1$ ;  $x_1 = 1, x_2 = 0$ ;  $x_1 = 1, x_2 = 1$ ). En general, para  $n$  variables tendremos  $2^n$  configuraciones mutuamente exclusivas. Como para cada una de estas configuraciones la función lógica puede asignarle un "1" ó un "0", tenemos  $2^{2^n}$  funciones lógicas posibles.

De las distintas formas de representar estas funciones vamos a comentar las más significativas desde el punto de vista electrónico.

1. *Forma normal disyuntiva* (suma de términos mínimos).
2. *Forma normal conjuntiva* (producto de términos máximos).
3. *Forma NAND*.
4. *Forma NOR*.

Por sencillez en la presentación y sin ninguna pérdida de generalidad nos referiremos en todos los casos a funciones de dos variables  $(x_1, x_2)$ .

### 1.4.1. Forma Normal Disyuntiva

La *forma normal disyuntiva* expresa una función lógica como suma de productos. Cada producto contiene a todas las variables, negadas o sin negar, sin repetirse ninguna. A los productos se les llama *términos mínimos* ("*minterms*") porque ocupan áreas mínimas (intersecciones) en los diagramas de Venn y se corresponden con las filas de la tabla de verdad, que a su vez son las distintas

configuraciones de valores posibles en las variables de entrada, interpretadas asociando al valor "0" la variable complementada y al valor "1" la variable sin complementar. Así, la configuración 0 1 se asocia al término mínimo  $\bar{x}_1 x_2$  porque si hacemos  $x_1 = 0$  y  $x_2 = 1$  sólo ese producto tendrá el valor 1 ( $\bar{x}_1 x_2 = \bar{0} \cdot 1 = 1 \cdot 1 = 1$ ). Todos los demás productos ( $\bar{x}_1 \bar{x}_2$ ,  $x_1 \bar{x}_2$  y  $x_1 x_2$ ) son cero para esos valores de las variables de entrada ( $x_1 = 0$  y  $x_2 = 1$ ). Sea cual fuere el valor de la configuración de entrada, sólo hay un término mínimo distinto de cero. Como estas configuraciones son mutuamente exclusivas, toda función lógica puede expresarse como *suma lógica de las configuraciones que la activan*. Así, para dos variables tenemos,

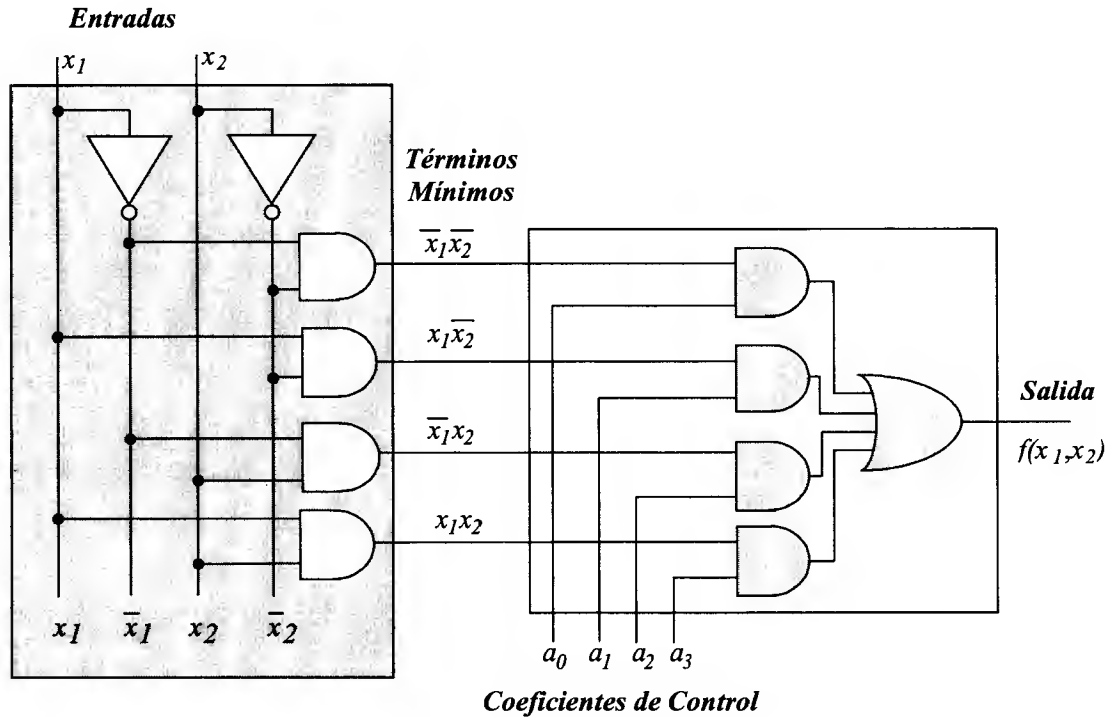
$$\begin{aligned} f_m(x_1, x_2) &= f(0,0)\bar{x}_1\bar{x}_2 + f(0,1)\bar{x}_1x_2 + f(1,0)x_1\bar{x}_2 + f(1,1)x_1x_2 = \\ &= a_0\bar{x}_1\bar{x}_2 + a_1\bar{x}_1x_2 + a_2x_1\bar{x}_2 + a_3x_1x_2 = \sum_{i=0}^{2^n-1} a_i m_i \end{aligned} \quad [1.12]$$

siendo  $a_i \in \{0,1\}$  y  $m_i = x_1^p x_2^q$ ,  $i = (p,q)$   $(p,q) \in \{0,1\}$

con la notación de Gilstrap ( $x^0 = \bar{x}$ ,  $x^1 = x$ )

Cada forma de representar una función lógica sugiere su procedimiento de síntesis. Así, en la forma normal disyuntiva que hemos visto aparecen productos de todas las variables ( $x_1, x_2$ ) y sus negadas ( $\bar{x}_1, \bar{x}_2$ ) para formar los cuatro términos mínimos ( $m_0 = \bar{x}_1 \bar{x}_2$ ,  $m_1 = \bar{x}_1 x_2$ ,  $m_2 = x_1 \bar{x}_2$  y  $m_3 = x_1 x_2$ ). Después estos términos mínimos se tienen que multiplicar por los coeficientes ( $a_0, a_1, a_2, a_3$ ) que determinan si van a pertenecer o no a una función concreta. Finalmente, los nuevos productos ( $a_i m_i$ ) se suman lógicamente. Si vamos construyendo el circuito que representa los procesos que hemos descrito tenemos la realización física de la expresión general de las funciones lógicas de dos variables, tal como se ilustra en la *figura 1.16*. Obsérvese que esta es una *función lógica universal* de forma que al variar los valores de los coeficientes  $\{a_i\}$  desde (0000) hasta (1111) vamos obteniendo las 16 funciones lógicas de dos variables que se incluyen en la parte inferior de la figura. Véase también que el subíndice decimal ( $i$ ) de la función ( $f_i$ ) pasado a binario genera el valor de los coeficientes  $\{a_i\}$  que sintetizan la función. Así,  $f_1$  es la función AND porque 1 en binario es 0001 ( $a_0 = a_1 = a_2 = 0$ ,  $a_3 = 1$ ) lo que supone que la función sólo posee el término mínimo  $m_3$  (1 1), o sea el correspondiente a  $x_1 x_2$ . Análogamente  $f_9$  es la función coincidencia que sólo se activa cuando las dos entradas son iguales, porque el decimal 9 en binario es 1001, es decir  $a_0 = 1$ ,

$a_1 = a_2 = 0, a_3 = 1$ . Posee por consiguiente los términos mínimos  $m_0(\bar{x}_1 \bar{x}_2)$  y  $m_3(x_1 x_2)$ .



$x_1$	$x_2$	$m_i$	$a_i$	$f_0$	$f_1$	$f_2$	$f_3$	...	$f_9$	...	$f_{14}$	$f_{15}$
0	0	$m_0$	$a_0$	0	0	0	0	...	1	...	1	1
0	1	$m_1$	$a_1$	0	0	0	0	...	0	...	1	1
1	0	$m_2$	$a_2$	0	0	1	1	...	0	...	1	1
1	1	$m_3$	$a_3$	0	1	0	1	...	1	...	0	1

**Figura 1.16.** Función universal para dos variables ( $x_1, x_2$ ) en forma normal disyuntiva y valor de los coeficientes  $\{a_i\}$  que la proyectan en cada una de las 16 funciones posibles para dos variables

### 1.4.2. Forma Normal Conjuntiva

En la forma *normal disyuntiva* que acabamos de ver, las funciones lógicas se representan mediante una *suma de productos*, a los que llamábamos términos mínimos (*minterms*) porque al representarlos en diagrama de Venn daban lugar a áreas mínimas (intersecciones entre áreas asociadas a variables o a sus negadas). Esta representación era completa y quedaba caracterizada por un vector de coeficientes  $\{a_i\}$  que nos decían cuáles eran los términos mínimos,  $\{m_i\}$ , que participaban en una determinada función. Es decir aquellos cuya presencia en la entrada de un circuito lógico hacían que su salida fuera “1”. Finalmente, los términos mínimos se asociaban a filas de la tabla de verdad (a configuraciones de entrada mutuamente exclusivas) con la interpretación siguiente: Un cero corresponde a una variable negada y un uno a una variable sin negar. Así, (0 1) corresponde a  $\bar{x}_1 x_2$ , por ejemplo.

Vamos a ver ahora una representación dual, en la que cualquier función lógica puede representarse como *producto de sumas*, a las que llamamos términos máximos (*maxterms*) porque al representarlos en diagramas de Venn dan lugar a *áreas máximas* (uniones de las áreas correspondientes a variables individuales o a sus negadas). Veremos que esta representación también es completa y queda caracterizada por un vector de coeficientes,  $\{A_i\}$ , que nos dicen cuáles son los términos máximos,  $\{M_i\}$  que participan en una determinada función. Como ahora los coeficientes,  $A_i$ , entran sumando, son relevantes aquellos términos máximos,  $M_i$ , cuyo  $A_i$  correspondiente sea cero. La suma de cualquier término lógico con “1” es “1” y el producto de “1” por cualquier término reproduce ese término, de forma dual a como ocurría con los productos y sumas en la representación por minterms.

Veamos ahora cómo se construyen los maxterms a partir de las  $2^n$  filas de la tabla de verdad de una función de  $n$  variables lógicas. Por razones pedagógicas y sin pérdida de generalidad usaremos como ilustración el caso de dos variables,  $(x_1, x_2)$ . La tabla de la *figura 1.17* muestra las cuatro configuraciones posibles (00, 01, 10, 11), los maxterms asociados ( $M_0, M_1, M_2, M_3$ ) y los coeficientes que designan la presencia o ausencia de cada uno de esos maxterms en una función.

Obsérvese que cada maxterm representa a una única fila de la tabla de verdad y una variable de entrada se complementa cuando su valor en la configuración de salida es “1” y no se complementa cuando su valor en esa fila es “0”. Así, a (00) le corresponde el maxterm  $M_0 = x_1 + x_2$ , a (01) el  $M_1 = x_1 + \bar{x}_2$ , a (10) el  $M_2 = \bar{x}_1 + x_2$  y a (11) el  $M_3 = \bar{x}_1 + \bar{x}_2$ .

$x_1$	$x_2$	Maxterms	Coeficientes
0	0	$x_1 + x_2$	$M_0$
0	1	$x_1 + \bar{x}_2$	$M_1$
1	0	$\bar{x}_1 + x_2$	$M_2$
1	1	$\bar{x}_1 + \bar{x}_2$	$M_3$

**Figura 1.17.** Tabla de verdad correspondiente a los términos máximos (maxterms).

De nuevo en la definición de los maxterms vemos la dualidad con los minterms. Allí construíamos un minterm multiplicando las variables sin negar o negadas dependiendo que en la fila de la tabla de verdad apareciera un “1” ó un “0” y decíamos que al evaluar la expresión del minterm correspondiente a cada fila con los valores de esa fila obteníamos un uno.

Ahora, con los maxterms, al evaluar el valor de la expresión de un maxterm con los valores correspondientes a su fila, obtenemos un cero. Por ejemplo, el maxterm  $M_0 = x_1 + x_2$  sólo posee el valor cero si  $x_1=0$  y  $x_2=0$ . Análogamente,  $M_1 = x_1 + \bar{x}_2$  sólo será cero si  $x_1=0$  y  $x_2=1$ , y así sucesivamente.

Como las cuatro configuraciones lógicas de la tabla son mutuamente exclusivas, sea cual fuere la configuración de entrada sólo habrá un maxterm distinto de 1, de forma dual a como sólo existía un minterm distinto de cero.

Por consiguiente y como consecuencia de esta exclusividad mutua, toda función lógica puede expresarse como producto lógico de maxterms. Llamando  $A_i$  a los coeficientes correspondientes a cada  $M_i$ , tenemos:

$$f_M(x_1, x_2) = (A_0 + M_0)(A_1 + M_1)(A_2 + M_2)(A_3 + M_3) \quad [1.13]$$

que es generalizable de forma inmediata para funciones de  $n$  variables y  $2^n$  maxterms,

$$f_M(x_1, x_2) = \prod_{i=0}^{2^n-1} (A_i + M_i) \quad [1.14]$$

En esta representación calculamos primero el OR de  $(A_i, M_i)$  y después el AND de esas sumas,

de forma dual a como hacíamos en la representación por minterms donde primero calculábamos el AND de  $(a_i, m_i)$  y después el OR de esos productos.

Allí, en minterms, sólo nos quedaban los  $m_i$  cuyos coeficientes  $a_i$  eran unos. Aquí, en maxterms, sólo nos quedan los  $M_i$  cuyos coeficientes  $A_i$  son cero. Así, el valor final de un producto de sumas sólo será cero si y sólo si el valor de todos los maxterms que contiene es cero. De otra forma, su valor será uno.

Dado que el subíndice usado para representar a los coeficientes es el decimal correspondiente a la expresión binaria de la configuración correspondiente ( $M_0$  para 00,  $M_1$  para 01, etc...), podemos representar de forma más compacta la función, usando sólo los subíndices de los maxterms que contiene. Así, si tiene  $M_0, M_1$  y  $M_3$ , la función será

$$f_M(x_1, x_2) = \prod M(0, 1, 3) \quad [1.15]$$

siendo 0, 1 y 3 los decimales cuya representación binaria corresponde a las filas de la tabla para las que la función es cero.

Minterms y maxterms son dos representaciones alternativas (duales) de la tabla de verdad de una función lógica. Además, para una misma función, una forma de representación puede obtenerse de la otra por simples manipulaciones algebraicas usando los teoremas de DeMorgan, ya que cada maxterm (minterm) puede obtenerse complementando el minterm (maxterm) correspondiente.

$$\text{Así, a } m_0 = \overline{x_1} \cdot \overline{x_2} \text{ le corresponde } M_0 = \overline{m_0} = \overline{\overline{x_1} \cdot \overline{x_2}} = x_1 + x_2 \quad [1.16]$$

Análogamente,

$$M_1 = \overline{m_1} = \overline{x_1 \cdot x_2} = x_1 + \overline{x_2} \quad [1.17]$$

$$M_2 = \overline{m_2} = \overline{x_1 \cdot \overline{x_2}} = \overline{x_1} + x_2 \quad [1.18]$$

$$M_3 = \overline{m_3} = \overline{\overline{x_1} \cdot x_2} = x_1 + \overline{x_2} \quad [1.19]$$

La tabla de la *figura 1.18* resume las dos representaciones, por minterms y maxterms, para funciones de dos variables.



Formas Canónicas					
$x_1$	$x_2$	Minterms	Coeficientes	Maxterms	Coeficientes
0	0	$\overline{x_1} \cdot \overline{x_2} = m_0$	$a_0$	$x_1 + x_2 = M_0 = \overline{m_0}$	$A_0$
0	1	$\overline{x_1} \cdot x_2 = m_1$	$a_1$	$x_1 + \overline{x_2} = M_1 = \overline{m_1}$	$A_1$
1	0	$x_1 \cdot \overline{x_2} = m_2$	$a_2$	$\overline{x_1} + x_2 = M_2 = \overline{m_2}$	$A_2$
1	1	$x_1 \cdot x_2 = m_3$	$a_3$	$\overline{x_1} + \overline{x_2} = M_3 = \overline{m_3}$	$A_3$
<b>Función</b>		$f_m = \sum_i a_i \cdot m_i$ <p><math>a_i=1</math> si el minterm está presente  <math>a_i=0</math> si no está presente</p>		$F_M = \prod_i (A_i + M_i)$ <p><math>A_i=0</math> si el maxterm está presente  <math>A_i=1</math> si no está presente</p>	

**Figura. 1.18.** Minterms, maxterms, sus coeficientes y las formas canónicas correspondientes para funciones de dos variables,  $(x_1, x_2)$

**Ejercicio:** Obtener la representación por minterms y maxterms de las siguientes funciones de dos variables;

- a) Coincidencia
- b) Anticoincidencia (OR-exclusivo)
- c) NAND
- d) NOR

**Solución**

Escribamos la tabla de verdad para cada uno de los casos y en la representación por minterms nos quedamos con los unos y en la de maxterms con los ceros.

- a) La figura 1.19 muestra la tabla de verdad para la coincidencia. En ella se han marcado las configuraciones correspondientes al resultado por minterms y por maxterms.

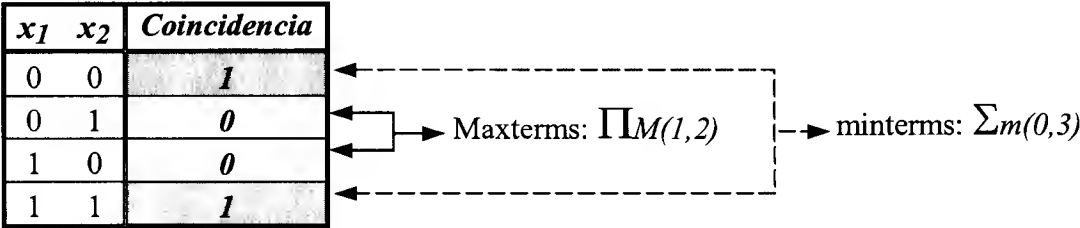


Figura 1.19. Tabla de verdad para la coincidencia

b), c) y d) La tabla de verdad para las funciones X-OR, NAND y NOR es la que se muestra en la figura 1.20 . Para representar estas funciones por minterms tomamos los términos mínimos que hacen 1 la función y para representar por maxterms la misma función tomamos los términos máximos que la hacen 0 (ver la última fila de la tabla).

$x_1$	$x_2$	X-OR	NAND	NOR
0	0	0	1	1
0	1	1	1	0
1	0	1	1	0
1	1	0	0	0
Representación por		⇓	⇓	⇓
minterms		$\sum m(1,2)$	$\sum m(0,1,2)$	$\sum m(0)$
Maxterms		$\prod M(0,3)$	$\prod M(3)$	$\prod M(1,2,3)$

Figura 1.20. Representación por minterms y por maxterms de las funciones X-OR, NAND y NOR.

Hay pues un procedimiento sencillo para pasar de una representación por maxterms (minterms) a otra por minterms (maxterms) de la misma función:

1. Me quedo con los números correspondientes a las configuraciones de entrada donde la función es uno. Así obtengo la representación por minterms de esa función.

2. Me quedo con la lista de los números donde la función es *cero*. Así obtengo su representación por maxterms.
3. Uso las expresiones correspondientes a ambas representaciones y compruebo, usando los teoremas y postulados de álgebra de Boole, que ambas representaciones coinciden. Así, para los cuatro ejemplos del ejercicio anterior tendríamos:

Caso a) minterms (0 y 3) y maxterms (1,2):

$$f_m^a = m_0 + m_3 = \bar{x}_1 \bar{x}_2 + x_1 x_2 \quad [1.20]$$

$$f_M^a = M_1 + M_2 = (x_1 + \bar{x}_2)(\bar{x}_1 + x_2) \quad [1.21]$$

desarrollando  $f_M^a$ , tenemos:

$$(x_1 + \bar{x}_2)(\bar{x}_1 + x_2) = \cancel{x_1 \bar{x}_1}^0 + x_1 x_2 + \bar{x}_2 \bar{x}_1 + \cancel{x_2 x_2}^0 = x_1 x_2 + \bar{x}_2 \bar{x}_1 \quad [1.22]$$

que es la misma expresión obtenida con minterms.

**Ejercicio.** *Comprobar ahora que en los casos b, c y d también se trata de la misma función desarrollando la expresión por maxterms y simplificándola.*

**Solución**

$$\text{Caso b)} \quad f_m^b = \bar{x}_1 x_2 + x_1 \bar{x}_2 \quad [1.23]$$

$$f_M^b = (x_1 + x_2)(\bar{x}_1 + \bar{x}_2) = x_1 \bar{x}_1 + x_1 \bar{x}_2 + x_2 \bar{x}_1 + x_2 \bar{x}_2 = x_1 \bar{x}_2 + x_2 \bar{x}_1 \quad [1.24]$$

$$\text{Caso c)} \quad f_m^c = \bar{x}_1 \bar{x}_2 + \bar{x}_1 x_2 + x_1 \bar{x}_2 = \bar{x}_1(\bar{x}_2 + x_2) + \bar{x}_2(x_1 + x_1) = \bar{x}_1 + \bar{x}_2 \quad [1.25]$$

$$f_M^c = (\bar{x}_1 + \bar{x}_2) \quad [1.26]$$

$$\text{Caso d)} \quad f_m^d = \bar{x}_1 \bar{x}_2 \quad [1.27]$$

$$f_M^d = (x_1 + \bar{x}_2)(\bar{x}_1 + x_2)(\bar{x}_1 + \bar{x}_2) = (x_1 x_2 + \bar{x}_1 \bar{x}_2)(\bar{x}_1 + \bar{x}_2) = \bar{x}_1 \bar{x}_2 \quad [1.28]$$

La expresión de cada maxterm,  $M_j$ , se obtiene de complementar la expresión del minterm de igual subíndice. Es decir,

$$M_j = \bar{m}_j \quad \text{y} \quad m_j = \bar{M}_j \quad \text{para} \quad i = 0, 1, \dots, 2^{n-1} \quad [1.29]$$

lo que es fácil de demostrar usando dos veces los teoremas de DeMorgan. Esto nos lleva a la relación global entre las dos formas de representación para una misma función,  $f$ :

$$\text{forma minterm de } f = \overline{\text{forma maxterm de } f} \quad [1.30]$$

$$\text{forma maxterm de } f = \overline{\text{forma minterm de } f} \quad [1.31]$$

El que aparezca  $\bar{f}$  en vez de  $f$  en estas expresiones es lo equivalente a lo que hicimos antes al quedarnos con los ceros de  $f$  para obtener los maxterms.

**Ejercicios (I)** Dadas las siguientes funciones de tres variables en representación por minterms, obtener su representación por maxterms

a)  $f_m = \sum m(0, 1, 2, 6, 7)$

b)  $f_m = \sum m(2, 3, 4, 5, 6)$

### **Solución**

- a) La expresión en minterms de  $f$ , ( $f_m$ ), nos da los “1” de  $f$ . Busquemos entonces sus ceros que estarán en las líneas complementarias de la tabla de verdad, es decir en (3,4,5). Por consiguiente, podremos escribir directamente que la expresión maxterms de  $f$  es:

$$f_M = M_3 M_4 M_5 \quad [1.32]$$

También podríamos haber usado la expresión general [1.31] que nos dice que la forma maxterm de  $f$ , ( $f_M$ ), es el complemento de la forma minterm de  $\bar{f}$ , ( $\bar{f}_m$ ). Así, el proceso a seguir es:

1) Dado  $f_m = \sum m(0, 1, 2, 6, 7)$  obtener la forma minterm de  $\bar{f}$ ,  $\bar{f}_m = \sum m(3, 4, 5)$ .

2) Complementarla para obtener la forma maxterm,

$$f_M = \overline{f_m} = \overline{\Sigma m(3,4,5)} = \overline{m_3} \overline{m_4} \overline{m_5} = M_3 M_4 M_5$$

b)  $f_m = \Sigma m(2,3,4,5,6)$

$$\overline{f_m} = \Sigma m(0,1,7)$$

$$f_M = \overline{f_m} = \overline{\Sigma m(0,1,7)} = \overline{m_0} \overline{m_1} \overline{m_7} = M_0 M_1 M_7 \quad [1.33]$$

**Ejercicios (II)** Dadas las siguientes funciones  $f_M$  en representación por maxterms, obtener sus representaciones equivalentes usando minterms,  $f_m$ .

a)  $f_M = M_2 M_3 M_4 M_5 M_6$

b)  $f_M = M_0 M_1 M_2 M_6 M_7$

### Solución

De forma dual a como razonamos en el ejercicio anterior, la expresión en maxterms de  $f$ , ( $f_M$ ), nos da los ceros de  $f$ . Busquemos entonces sus unos que estarán en las líneas complementarias de la tabla de verdad, es decir en (0, 1, 7). Por consiguiente, podemos escribir directamente que la expresión en minterm de  $f$ , ( $f_m$ ), será:

$$f_m = m_0 + m_1 + m_7 \quad [1.34]$$

También podríamos haber usado la expresión general [1.30] que nos dice que la forma minterm de  $f$ , ( $f_m$ ), es el complemento de la forma maxterm de  $\overline{f}$ . Así, los pasos a seguir son:

1) Dado  $f_M = \prod M(2,3,4,5,6)$ , obtener la forma maxterm de  $\overline{f}$ ,  $\overline{f_M} = \prod M(0,1,7)$ .

2) Complementarla

$$f_m = \overline{f_M} = \overline{\prod M(0,1,7)} = \overline{M_0} + \overline{M_1} + \overline{M_7} = m_0 + m_1 + m_7$$

Análogamente razonaríamos en el caso b):

$$f_m = \overline{f_M} = \overline{\prod M(3,4,5)} = \overline{M_3} + \overline{M_4} + \overline{M_5} = m_3 + m_4 + m_5 \quad [1.35]$$

Conviene recordar que siempre que tengamos dudas sobre si nuestra conversión de minterms a maxterms o viceversa, es o no es correcta, podemos pasar a la representación en términos de las

variables reales,  $x_1$ ,  $x_2$ ,  $x_3$ , donde ambas representaciones deben coincidir, como vimos en las ecuaciones [1.20] a [1.28] del ejercicio anterior.

---

## 1.5. OTRAS REPRESENTACIONES COMPLETAS (NAND, NOR)

Tanto en la suma de productos como en el producto de sumas, para sintetizar cualquier función sólo necesitamos la realización electrónica de tres operadores: **AND**, **OR** y **NO**. Por eso decimos que estos tres operadores constituyen un conjunto completo. Existen, sin embargo otros conjuntos completos más pequeños y, por consiguiente, más útiles a la hora de realizarlos físicamente. Nos referimos a los circuitos **NAND** (AND seguido de inversor) y **NOR** (OR seguido de inversor). Con cualquiera de ellos se puede sintetizar cualquier función lógica. Nos vamos acercando así al objetivo de este tema que es definir las exigencias computacionales mínimas del tratamiento digital de la información. Digámoslo ya avanzando conocimientos. El propósito de la Electrónica Digital es obtener realizaciones físicas en tecnología bipolar o unipolar de *conjuntos completos de operadores lógicos*. Con ellos podremos sintetizar cualquier función *combinacional* (de codificación aritmético-lógica o de ruta de datos) y con leves modificaciones, cualquier función *secuencial*. Razones adicionales sobre facilidad de fabricación harán que sea ventajosa la representación mediante un único operador, **NAND** ó **NOR**. Así, todo lo que necesitamos es una *suma* o un *producto* seguidos de un *inversor*.

La *figura 1.21* incluye las tablas de verdad, los símbolos y las expresiones de los operadores **NAND** y **NOR**. Como muestra de su complitud se ilustra también en la *figura 1.22* la síntesis de los operadores del anterior conjunto completo (AND, OR y complemento) en términos de sólo operadores **NAND** y sólo operadores **NOR**.

De forma general, aplicando los teoremas de DeMorgan a la forma normal disyuntiva, podemos obtener las expresiones equivalentes en **NAND** y **NOR**. Veamos el caso **NAND**:

$$f = \sum_i a_i \cdot m_i \quad [1.36]$$

Si complementamos ahora dos veces obtenemos:

$$f = \overline{\overline{\sum_i (a_i m_i)}} = \overline{\prod_i (\overline{a_i m_i})} \quad [1.37]$$

Con lo que ya tenemos la expresión de  $f$  usando dos niveles de puertas NAND. En el primer nivel obtenemos el NAND de  $a_i$  y  $m_i$ . En un segundo nivel obtenemos el NAND de las expresiones anteriores. Si llamamos  $P_i$  a la negación de los productos  $a_i m_i$  (su NAND) podemos volver a escribir la expresión anterior de la forma:

$$f = \overline{\overline{\sum_i (a_i m_i)}} = \overline{\prod_i (\overline{a_i m_i})} = \overline{\prod_i P_i} = (a_0 \uparrow m_0) \uparrow (a_1 \uparrow m_1) \uparrow \dots \uparrow (a_{2^n-1} \uparrow m_{2^n-1}) \quad [1.38]$$

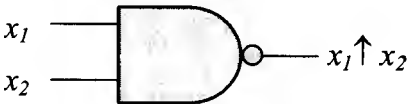
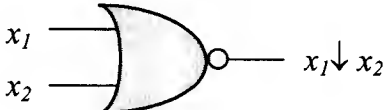
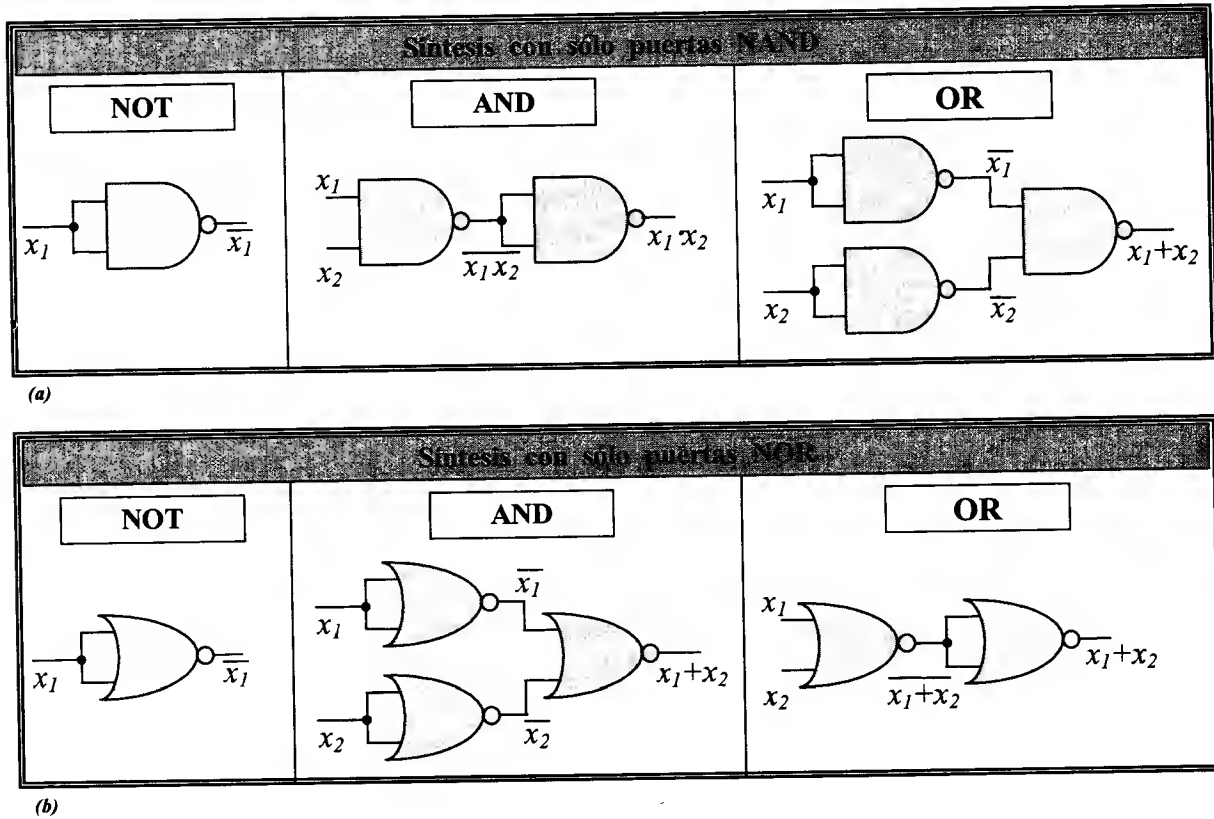
NAND ( $\uparrow$ )	NOR ( $\downarrow$ )																														
																															
<table border="1"> <tr> <th><math>x_1</math></th> <th><math>x_2</math></th> <th><math>\uparrow</math></th> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table>	$x_1$	$x_2$	$\uparrow$	0	0	1	0	1	1	1	0	1	1	1	0	<table border="1"> <tr> <th><math>x_1</math></th> <th><math>x_2</math></th> <th><math>\downarrow</math></th> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table>	$x_1$	$x_2$	$\downarrow$	0	0	1	0	1	0	1	0	0	1	1	0
$x_1$	$x_2$	$\uparrow$																													
0	0	1																													
0	1	1																													
1	0	1																													
1	1	0																													
$x_1$	$x_2$	$\downarrow$																													
0	0	1																													
0	1	0																													
1	0	0																													
1	1	0																													
$x_1 \uparrow x_2 = \overline{\overline{x_1} \overline{x_2}} + \overline{\overline{x_1} x_2} + \overline{x_1 \overline{x_2}} = \overline{x_1 x_2} = \overline{x_1} + \overline{x_2}$	$x_1 \downarrow x_2 = \overline{x_1 x_2} = \overline{x_1} + \overline{x_2}$																														

Figura 1.21. Definición de los operadores NAND y NOR.



**Figura 1.22.** Síntesis del conjunto completo (AND, OR, NOT). (a) Usando sólo operadores NAND. (b) Usando sólo operadores NOR.

Por ejemplo, para  $f = x_1 + x_2$  tendremos:

$$\overline{f} = \overline{x_1 + x_2} = \overline{x_1} \cdot \overline{x_2} \quad [1.39]$$

$$\overline{\overline{f}} = \overline{\overline{x_1} \cdot \overline{x_2}} = f \quad [1.40]$$

donde  $\overline{x_1}$  es el NAND de  $x_1$  consigo misma,  $\overline{x_2}$  el NAND de  $x_2$  consigo misma y, finalmente un nuevo NAND con los resultados anteriores. El circuito correspondiente es el que se ilustra en la figura 1.22. Análogamente obtendríamos el correspondiente al producto  $x_1 \cdot x_2$ .

$$f = x_1 \cdot x_2 \quad [1.41]$$



$$\overline{\overline{f}} = \overline{\overline{x_1 \cdot x_2}} \quad [1.42]$$

Ahora que hemos visto de forma individual los distintos tipos de representación es hora de comprobar con algunos ejemplos lo que hemos aprendido.

**Ejercicio:** Representar con sólo puertas NAND la siguiente función:

$$A = XY\overline{Z} + XYZ + X(Y + \overline{Z})$$

Para el paso a NAND podemos usar el siguiente procedimiento general:

1. **Obtener una expresión mínima** en forma de suma de productos (este apartado lo veremos al comentar el **objetivo 6**).
2. **Complementar dos veces.** Al complementar dos veces hemos dejado la función como estaba, por lo que siempre podremos hacerlo.
3. **Aplicar los teoremas de De Morgan.** La aplicación repetida de los Teoremas de De Morgan debe pararse cuando:
  - a) **Sólo encontramos variables negadas** ( $\overline{X}, \overline{Y}$ ) que se sintetizan con un inversor, el cual es un caso particular de una puerta NAND en la que se unen las dos entradas.
  - b) **Sólo encontramos negaciones de productos** ( $\overline{X\overline{Y}}, \overline{\overline{X}Y}, \dots$ ).

**Solución:**

De acuerdo con el procedimiento descrito, primero complementamos dos veces. Así,

$$A = \overline{\overline{A}} = \overline{\overline{XY\overline{Z} + XYZ + X(Y + \overline{Z})}} = \overline{\overline{XY\overline{Z}} + \overline{\overline{XYZ}} + \overline{\overline{X(Y + \overline{Z})}}} \quad [1.43]$$

A continuación aplicaremos los teoremas de DeMorgan, resultando:

$$A = \overline{\overline{XY\overline{Z}} + \overline{\overline{XYZ}} + \overline{\overline{X(Y + \overline{Z})}}} = \overline{\overline{XY\overline{Z}}} \cdot \overline{\overline{\overline{XYZ}}} \cdot \overline{\overline{\overline{X(Y + \overline{Z})}}} \quad [1.44]$$

Por último, comprobaremos con la tabla de verdad que ambas representaciones [1.43] y [1.44] coinciden.

$X$	$Y$	$Z$	$XY\bar{Z}$	$XYZ$	$X(Y + \bar{Z})$	$A [1.43]$	$\overline{XY\bar{Z}}$	$\overline{XYZ}$	$\overline{XY}$	$\overline{X\bar{Z}}$	$A [1.44]$
0	0	0	0	0	0	0	1	1	1	1	0
0	0	1	0	0	0	0	1	1	1	1	0
0	1	0	0	0	0	0	1	1	1	1	0
0	1	1	0	0	0	0	1	1	1	1	0
1	0	0	0	0	1	1	1	1	1	0	1
1	0	1	0	0	0	0	1	1	1	1	0
1	1	0	1	0	1	1	0	1	0	0	1
1	1	1	0	1	1	1	1	0	0	1	1

Figura 1.23. Tabla de verdad para demostrar que las expresiones [1.43] y [1.44] coinciden.

**Ejercicio:** Pasar de NAND a NOR la siguiente función:

$$A = f(X, Y, Z) = \overline{X} \bar{Y} Z + X Y Z$$

### Solución

Para el *paso de NAND a NOR*, también es fácil comprobar el grado de comprensión alcanzado usando el siguiente procedimiento general que describimos sobre un ejemplo:

1. Partimos de una expresión que sólo usa puertas NAND, por ejemplo

$$A = f(X, Y, Z) = \overline{X} \bar{Y} Z + X Y Z \quad [1.45]$$

2. Aplicamos DeMorgan:  $A = \overline{\overline{X} \bar{Y} Z + X Y Z} = (X + Y + \bar{Z})(\bar{X} + \bar{Y} + \bar{Z}) \quad [1.46]$

3. Dibujamos el circuito correspondiente en la lógica de dos niveles (OR-AND)

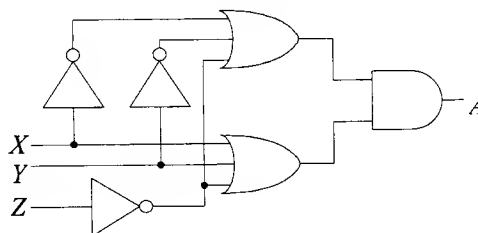


Figura 1.24. Implementación de la función  $A$  con inversores y puertas AND y OR

4. Negamos dos veces y aplicamos de nuevo DeMorgan para obtener la representación con *sólo* puertas NOR

$$A = \overline{\overline{(X + Y + \overline{Z})}(\overline{\overline{X}} + \overline{\overline{Y}} + \overline{\overline{Z}})} = \overline{(X + Y + \overline{Z}) + (\overline{X} + \overline{Y} + \overline{Z})} \quad [1.47]$$

5. Representamos el circuito correspondiente:

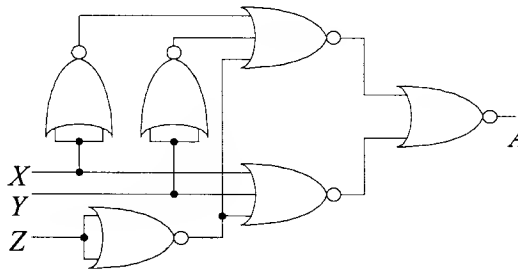


Figura 1.25. Implementación de la función A con sólo puertas NOR

6. Usamos el simulador para comprobar que *ambos circuitos se comportan igual*

Dado que en la versión reducida y de uso libre del simulador PSpice no contamos con puertas OR de 3 entradas hemos usado las de 2 entradas, realizando las funciones OR de tres variable en dos etapas. En una primera etapa hacemos la función OR de dos variables y en la siguiente etapa hacemos la función OR de la tercera variable con el resultado de la primera etapa.

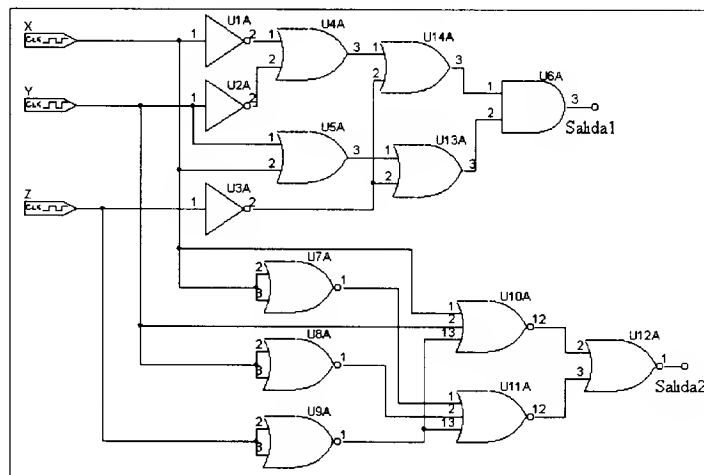
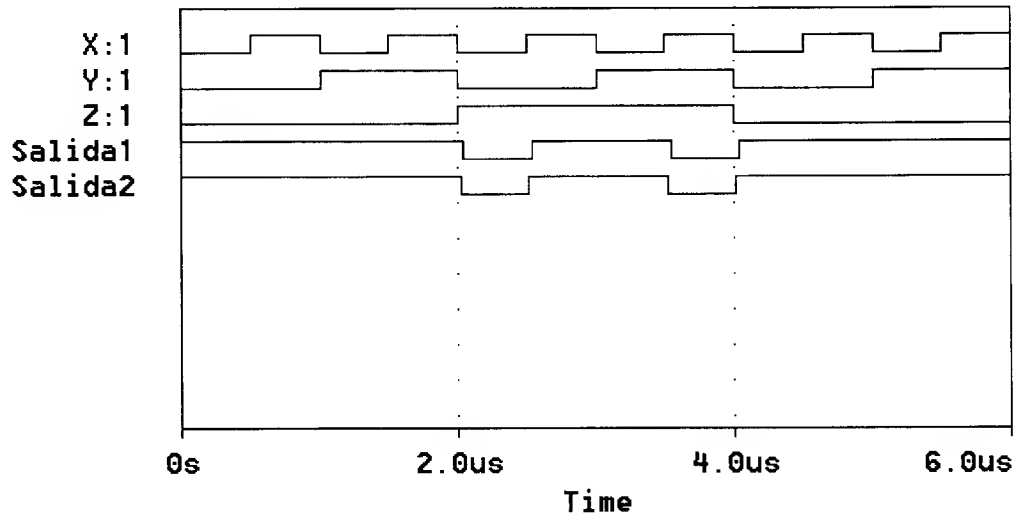


Figura 1.26. Implementación de la función A con el simulador PSpice.

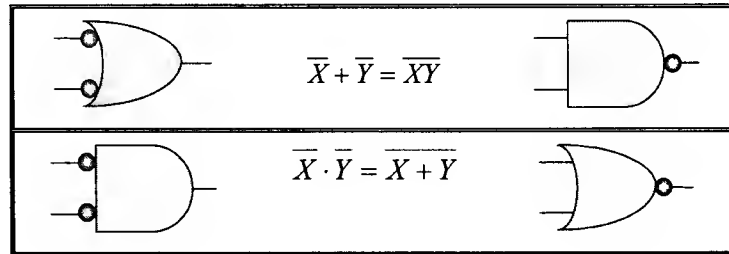


**Figura 1.27** Resultado de la simulación y comprobación de la validez de ambos diseños.

Como puede observarse en el cronograma las dos salidas coinciden para las ocho configuraciones de entrada.

Una forma más sencilla y rápida de conseguir la transformación, en los casos en los que disponemos del circuito en su representación OR-AND, es emplear un método gráfico, consistente en *negar las salidas de las puertas del primer nivel y las entradas de las puertas del segundo nivel*. Con esta doble negación, que no altera lo más mínimo la función del circuito, se convierten las puertas AND (OR) del primer nivel en puertas NAND (NOR) y las puertas OR (AND) del segundo nivel en puertas OR (AND) con las entradas negadas, o lo que es lo mismo en puertas NAND (NOR) con las entradas sin negar (*figura 1.28*). Estos cambios, no son más que el resultado de la aplicación directa de la representación gráfica de los teoremas de DeMorgan.

Este método es igualmente válido para obtener la representación sólo con puertas NAND partiendo del esquema del circuito representado con la lógica de dos niveles AND-OR (suma de minterms), como para obtener la representación sólo con puertas NOR a partir del esquema del circuito representado con la lógica de dos niveles OR-AND (producto de maxterms)

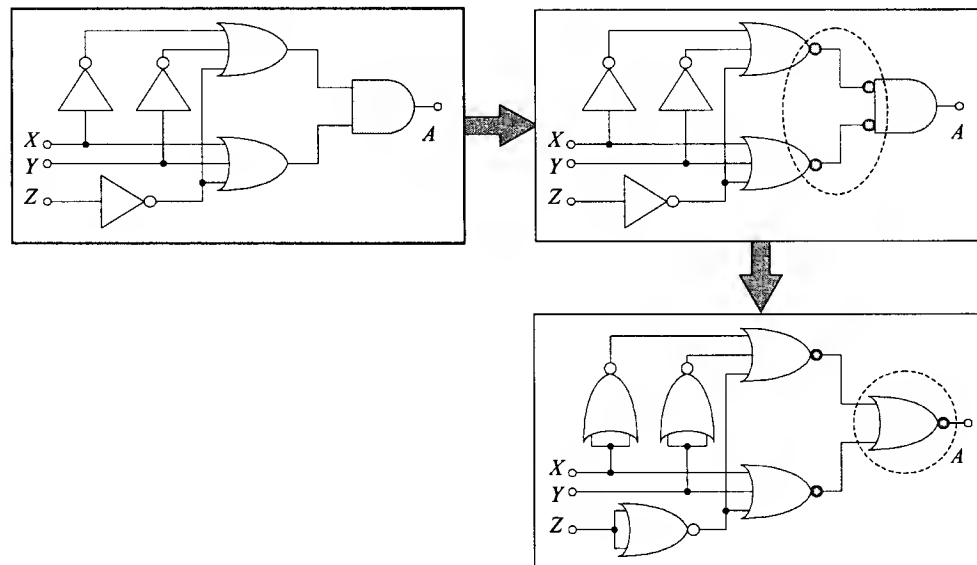


**Figura 1.28.** Implementación de los teoremas de DeMorgan

**Ejercicio:** Aplicar el método gráfico descrito anteriormente al circuito del ejercicio anterior:

**Solución**

La figura 1.29 muestra los pasos sucesivos para pasar un circuito implementado con inversores, puertas OR y AND a su implementación con sólo puertas NOR. Para ello, primero hemos aplicado el método gráfico para el paso de OR-AND a sólo NOR y después hemos sustituido los inversores por puertas NOR con las entradas unidas.



**Figura 1.29.** Ilustración del método gráfico.

## 1.6. ANÁLISIS Y SÍNTESIS

Vamos a ver de forma sencilla y a través de ejemplos en que consisten los procesos de *análisis* y *síntesis* de circuitos lógicos.

*Analizar un circuito lógico es encontrar la función lógica que calcula, a partir del esquema de conexión de las variables de entrada con los distintos operadores hasta llegar a la variable de salida.*

La variable o variables dependientes (la función) se encuentra a la salida de la última puerta y, por consiguiente, para obtener la expresión lógica que la conecta con las entradas basta seguir el camino de las variables de entrada hasta llegar a la salida y comprobar por inducción completa que ambos (circuito y expresión lógica) coinciden porque poseen la misma tabla de verdad. Los problemas E.1.1 y E.1.2 resueltos en el texto de problemas son representativos de este procedimiento.

Aquí, en el primer tema dedicado a los circuitos lógicos combinacionales de pocas variables, el problema de análisis es realmente sencillo. Veremos más adelante, al estudiar circuitos secuenciales, cómo no es tan sencillo “*descubrir*” la función global que realizan a partir de las conexiones entre sus componentes. La clave está en el concepto de “estado” y en la existencia de lazos de realimentación. Aprovechémonos de momento de la sencillez del análisis en lógica combinatorial, obteniendo la función lógica que realizan los circuitos de los dos ejercicios siguientes en los que hemos ido señalando las distintas transformaciones que sufren las variables de entrada,  $A$ ,  $B$  y  $C$  hasta llegar a las funciones  $F_1$ ,  $F_2$  y  $F_3$ .

---

**Ejercicio:** *Analizar el circuito de la figura corresponde a una función múltiple ( $F_1$ ,  $F_2$ ) de dos variables ( $A$  y  $B$ ).*

***Solución:***

La forma de analizar un circuito es seguir el camino de la señal desde las entradas,  $A$  y  $B$ , hasta las salidas,  $F_1$  y  $F_2$ , anotando las transformaciones que introducen los operadores que se encuentran en el camino.

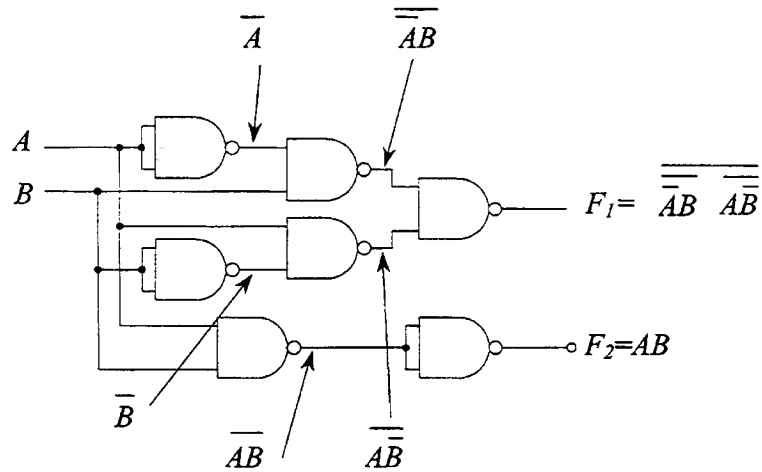


Figura 1.30.

**Ejercicio:** Repetir el procedimiento de análisis para el circuito de la figura 1.31.

**Solución:**

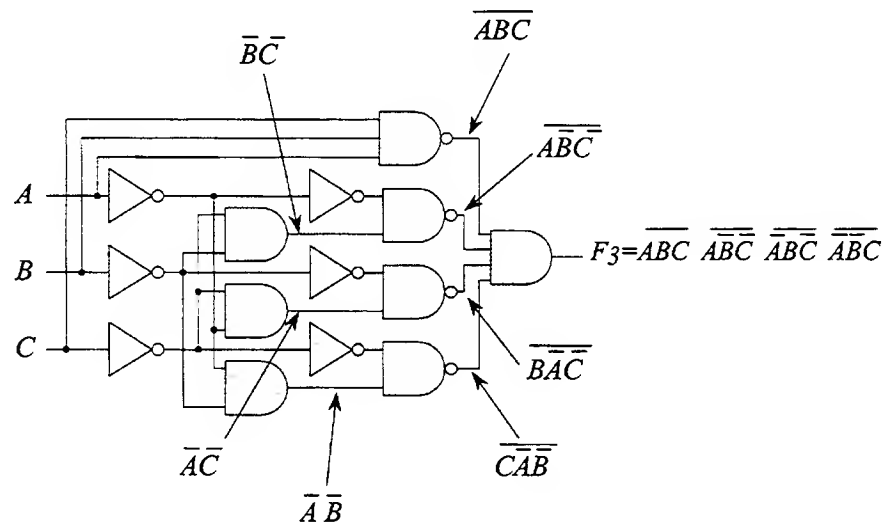


Figura 1.31.

Las dos fases esenciales en todo proceso de *síntesis* son:

**Fase a:** Paso de una descripción en lenguaje natural a una función lógica.

**Fase b:** Paso de una función lógica a su circuito correspondiente.

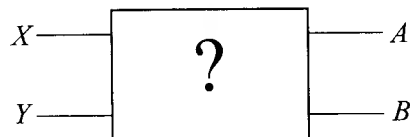
Veamos ahora algunos ejemplos de ambas fases.

**Ejercicio:** Obtener una descripción lógica, incluyendo la selección de variables de entrada y salida, de un circuito con dos entradas y dos salidas tal que la primera salida esté en alta cuando el valor en las dos entradas es el mismo y la segunda salida está en alta cuando no coinciden las entradas. Obtener también el circuito que implementa dichas funciones

**Solución:**

**Fase a:** Paso de una descripción en lenguaje natural a una función lógica.

1. Empezamos asignando dos variables ( $X, Y$ ) a las entradas y otras dos ( $A, B$ ) a las salidas.



2. Construimos una tabla de verdad en la que describimos las cuatro configuraciones lógicas posibles en la entrada y ponemos un uno en los términos mínimos que hacen pasar  $A$  a alta y en otra columna, los que hacen pasar  $B$  a alta.

$X$	$Y$	$A$	$B$
0	0	1	0
0	1	0	1
1	0	0	1
1	1	1	0

**Figura 1.32.** Tabla de verdad



Si hacemos la síntesis usando los términos mínimos tomamos las configuraciones de entrada que hacen que la salida sea 1. Así, para la función  $A$ , tenemos:

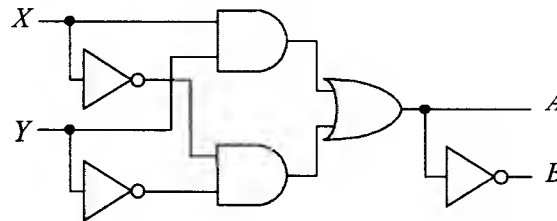
$$A = \overline{X} \overline{Y} + X Y \quad [1.48]$$

$$\text{Análogamente para la salida obtenemos: } B = \overline{X} Y + X \overline{Y} \quad [1.49]$$

Es decir, la función  $A$  es la *coincidencia* y la  $B$  la *anticoincidencia* ó OR exclusivo.

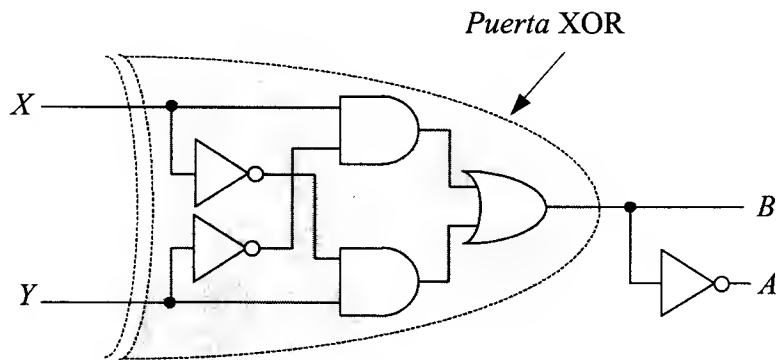
$$\text{Así, } A = \overline{X \oplus Y} \quad \text{y} \quad B = X \oplus Y \quad [1.50]$$

**Fase b:** Paso de una función lógica a su circuito correspondiente.



**Figura 1.33.**

Como puede observarse esta no es la única forma de implementar estas funciones. En este caso hemos hecho el circuito para la función  $A$  y después, negando la salida, hemos obtenido la función  $B$ . De igual forma, es correcta la implementación si hacemos el circuito que implementa la función  $B$  y después complementando su salida obtenemos la función  $A$ . Así:



**Figura 1.34.**

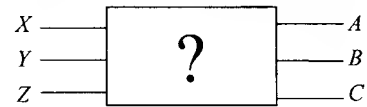
**Ejercicio:** Diseñar y minimizar un circuito lógico de tres variables de entrada y tres de salida de forma que responda a cada una de las siguientes condiciones:

- a) Dos o más de las entradas estén en alta a la vez.
- b) El decimal equivalente al número binario que representan las tres entradas sea impar (1, 3, 5 ó 7).
- c) El decimal equivalente al número binario que representan las tres entradas sea par (0, 2, 4 ó 6).

**Solución:**

**Fase a:** Paso de una descripción en lenguaje natural a una función lógica.

1. Empezamos asignando tres variables (*X*, *Y*, *Z*) a las entradas y otras tres (*A*, *B*, *C*) a las salidas.



2. Construimos la tabla de verdad. Para ello en la primera columna ponemos los números decimales de 0 a 7, ya que con tres variables podremos representar  $2^3 = 8$  configuraciones de entrada. Las tres columnas siguientes son las entradas y las siguientes columnas son las tres salidas que nos piden en el problema y que se corresponden con las soluciones a los apartados *a*, *b* y *c* del enunciado. Así, la tabla de verdad es:

Nº Decimal	X Y Z	Dos o mas entradas en Alta A	Nº decimal impar B	Nº decimal par C
0	0 0 0	0	0	1
1	0 0 1	0	1	0
2	0 1 0	0	0	1
3	0 1 1	1	1	0
4	1 0 0	0	0	1
5	1 0 1	1	1	0
6	1 1 0	1	0	1
7	1 1 1	1	1	0

Figura 1.35.

Ahora vemos cuáles son las configuraciones de entrada que producen un 1 en las distintas salidas. Es decir, vemos qué términos mínimos participan en la producción de cada salida. Así,

$$A = \bar{X}YZ + X\bar{Y}Z + XY\bar{Z} + XYZ \quad [1.51]$$

$$B = \bar{X}\bar{Y}Z + \bar{X}YZ + X\bar{Y}Z + XYZ \quad [1.52]$$

$$C = \bar{X}\bar{Y}\bar{Z} + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + XYZ \quad [1.53]$$

Aunque veremos más adelante la minimización por mapas de Karnaugh dentro del Objetivo 6, aquí utilizaremos la *aplicación directa de los teoremas y postulados del Algebra de Boole* para simplificar estos sencillos ejemplos de síntesis.

$$A = \bar{X}YZ + X\bar{Y}Z + XY\bar{Z} + XYZ = YZ(X + \bar{X}) + XY(Z + \bar{Z}) + XZ(Y + \bar{Y}) = YZ + XY + XZ \quad [1.54]$$

Análogamente:

$$B = \bar{X}\bar{Y}Z + \bar{X}YZ + X\bar{Y}Z + XYZ = \bar{Y}Z(X + \bar{X}) + YZ(X + \bar{X}) = \bar{Y}Z + YZ = Z(Y + \bar{Y}) = Z$$

$$C = \bar{X}\bar{Y}\bar{Z} + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + XYZ = \bar{Y}\bar{Z}(X + \bar{X}) + Y\bar{Z}(X + \bar{X}) = \bar{Y}\bar{Z} + Y\bar{Z} = \bar{Z}(Y + \bar{Y}) = \bar{Z}$$

En estos dos últimos casos podemos llegar a la solución por simple inspección de la tabla de verdad, ya que si comparamos la columna  $B$  con las de las variables de entrada observamos que coincide con la columna correspondiente a la entrada  $Z$ . Por tanto  $B=Z$ . Análogamente los valores de la salida  $C$  coincide con los valores complementados de la columna correspondiente a la entrada  $Z$ . Por tanto  $B = \bar{Z}$ . Sin embargo esto no suele ocurrir en la mayoría de los casos por lo que en general hay que seguir los pasos de alguno de los procedimientos de minimización.

**Fase b: Implementación.** Paso de una función lógica a su circuito correspondiente.

En esta fase, evidentemente, podemos usar cualquiera de las formas completas (AND, OR, NOT: sólo NAND o sólo NOR) que ya conocemos. Así, en el primer caso obtendríamos un circuito como el que se ilustra en la *figura 1.36*.

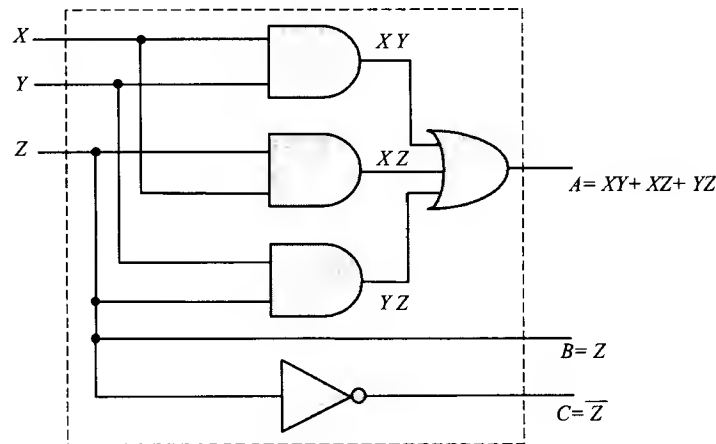


Figura 1.36.

## 1.7 INTRODUCCIÓN A LA MINIMIZACIÓN

*Minimizar una función lógica es obtener la expresión más simplificada posible para la misma de forma que el número de operadores necesarios para su síntesis es también mínimo.*

Un tratamiento adecuado de este problema queda fuera del alcance de este tema de introducción a la electrónica digital, por lo que sólo presentaremos los conceptos mínimos necesarios para comprender los procedimientos de síntesis de las funciones combinacionales y secuenciales que estudiaremos más adelante.

Si pensamos en términos de *suma de productos* parece intuitivo afirmar que una función está en su forma mínima cuando es mínimo el número de productos y, a igualdad de productos, cuando es mínimo el número de variables. Desde un punto de vista algebraico y para pocas variables, el uso directo de algunos postulados y teoremas del álgebra de Boole es suficiente para conseguir la minimización. Por ejemplo,

$$x_1 x_2 + x_1 \bar{x}_2 = x_1 (x_2 + \bar{x}_2) = x_1 \quad [1.55]$$

$$x_1 + x_1 x_2 = x_1 (x_2 + 1) = x_1 \quad [1.56]$$

Cada término mínimo posee dos vecinos de los que sólo le diferencia el estado de una variable que en un caso está complementada y en otro sin complementar de forma que si aparecen juntos en una expresión se puede sacar factor común y eliminar la variable que cambia. Los cuatro términos mínimos para dos variables son  $\bar{x}_1 \bar{x}_2$ ,  $\bar{x}_1 x_2$ ,  $x_1 \bar{x}_2$ ,  $x_1 x_2$ . Así, para el primer término,  $\bar{x}_1 \bar{x}_2$  tenemos dos vecinos:  $\bar{x}_1 x_2$  (cambia sólo  $x_2$ ), y  $x_1 \bar{x}_2$  (cambia sólo  $x_1$ ), con lo que podríamos obtener las siguientes simplificaciones:

$$\bar{x}_1 \bar{x}_2 + \bar{x}_1 x_2 = \bar{x}_1 (x_2 + \bar{x}_2) = \bar{x}_1 \quad [1.57]$$

$$\bar{x}_1 \bar{x}_2 + x_1 \bar{x}_2 = \bar{x}_2 (\bar{x}_1 + x_1) = \bar{x}_2 \quad [1.58]$$

Análogamente, los vecinos de  $x_1 x_2$  son ( $x_1 \bar{x}_2$  y  $\bar{x}_1 x_2$ ) y los de  $x_1 \bar{x}_2$  son ( $x_1 x_2$  y  $\bar{x}_1 \bar{x}_2$ ). Finalmente, los vecinos de  $\bar{x}_1 x_2$  son ( $\bar{x}_1 \bar{x}_2$  y  $x_1 x_2$ ).

Los teoremas más adecuados para la minimización son el de la *adyacencia* y el de *idempotencia*.

El de la *adyacencia* debido a que, siempre que se puede sacar factor común una variable (o el producto de varias) de la suma de otro término con su negado, ambos términos quedan reducido a la parte común. Es decir, los términos sólo se diferencian en un bit.

El de la *idempotencia* porque en muchas ocasiones se usa un mismo término con dos o más adyacentes para aplicar el teorema de la adyacencia.

Para pasar de  $\bar{X}YZ + \bar{X}Y\bar{Z}$  a  $\bar{X}Y$  se aplica el teorema de la adyacencia, ya que

$$\bar{X}YZ + \bar{X}Y\bar{Z} = \bar{X}Y(Z + \bar{Z}) = \bar{X}Y \quad \text{y} \quad Z + \bar{Z} = 1 \quad [1.59]$$

Como podemos observar estos dos términos sólo se diferencian en la variable  $Z$ , que toma un valor y su complementario.

Veamos ahora si lo hemos comprendido intentando resolver los siguientes ejercicios:

**Ejercicio:** Minimizar algebraicamente las siguientes funciones:

a)  $f = AB + C(AB + \bar{C})$

$$b) f = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + AB\overline{C} + A\overline{B}\overline{C}$$

**Solución:**

$$a) f = AB + C(AB + \overline{C}) = AB + CAB + C\overline{C} = AB(C + 1) = AB \quad [1.60]$$

$\begin{array}{cc} 0 \nearrow & 1 \nearrow \end{array}$

$$b) f = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + AB\overline{C} + A\overline{B}\overline{C} \quad [1.61]$$

$$\overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + AB\overline{C} + A\overline{B}\overline{C} = \overline{A}\overline{C}(B + \overline{B}) + A\overline{C}(B + \overline{B}) = \overline{A}\overline{C} + A\overline{C} = \overline{C}(A + \overline{A}) = \overline{C} \quad [1.62]$$

A veces es conveniente expandir primero una función para minimizarla después. En general, esta estrategia es recomendable cuando no es evidente que podamos conseguir una simplificación por aplicación directa de los teoremas de adyacencia. Por ejemplo,

$$f = \overline{X}Y + XY\overline{Z} + X\overline{Y} + \overline{X}\overline{Y}\overline{Z} \quad [1.63]$$

Introducimos la variable Z que falta en el primer y tercer término multiplicando por 1. Es decir, multiplicamos ambos términos por:  $Z + \overline{Z} = 1$

$$f = \overline{X}Y(Z + \overline{Z}) + XY\overline{Z} + X\overline{Y}(Z + \overline{Z}) + \overline{X}\overline{Y}\overline{Z} = \overline{X}YZ + \overline{X}Y\overline{Z} + XY\overline{Z} + X\overline{Y}Z + X\overline{Y}\overline{Z} + \overline{X}\overline{Y}\overline{Z} \quad [1.64]$$

Ahora agrupamos los términos que sólo se diferencian en una variable y aplicamos el teorema de Adyacencia

$$f = \overline{X}YZ + \overline{X}Y\overline{Z} + XY\overline{Z} + X\overline{Y}Z + X\overline{Y}\overline{Z} + \overline{X}\overline{Y}\overline{Z} = \overline{X}Y + Y\overline{Z} + X\overline{Y} + \overline{Y}\overline{Z} \quad [1.65]$$

Aplicando de nuevo el teorema de adyacencia

$$f = \overline{X}Y + Y\overline{Z} + X\overline{Y} + \overline{Y}\overline{Z} = \overline{X}Y + X\overline{Y} + \overline{Z} \quad [1.66]$$

Cuando aumenta el número de variables es difícil identificar estos pares de términos y de aquí nace el método de *Veitch-Karnaugh* (V-K) para minimizar funciones. Se trata en esencia de un método gráfico que distribuye sobre distintas áreas contiguas los términos mínimos de forma tal que los pares simplificables siempre son vecinos y el proceso de simplificación se puede realizar por simple inspección visual.

La figura 1.37 muestra los diagramas V-K para una, dos y tres variables. Para una variable el rectángulo posee dos áreas, una para  $x = 0$  y otra para  $x = 1$ . Para dos variables es un cuadrado con cuatro áreas, una para cada configuración (00, 01, 10, 11). Para tres variables es un rectángulo con ocho áreas (desde 000, hasta 111). Obsérvese que en todos los casos la diferencia entre dos cuadros consecutivos es sólo en una variable. Para el caso de 3 variables es necesario considerar al diagrama como una superficie cilíndrica de forma que sus extremos derecho e izquierdo están unidos. Una vez que conocemos el significado de cada rectángulo, dejémoslo vacío y pongamos sólo "1" en las áreas que corresponden a los términos mínimos que posee una función específica.

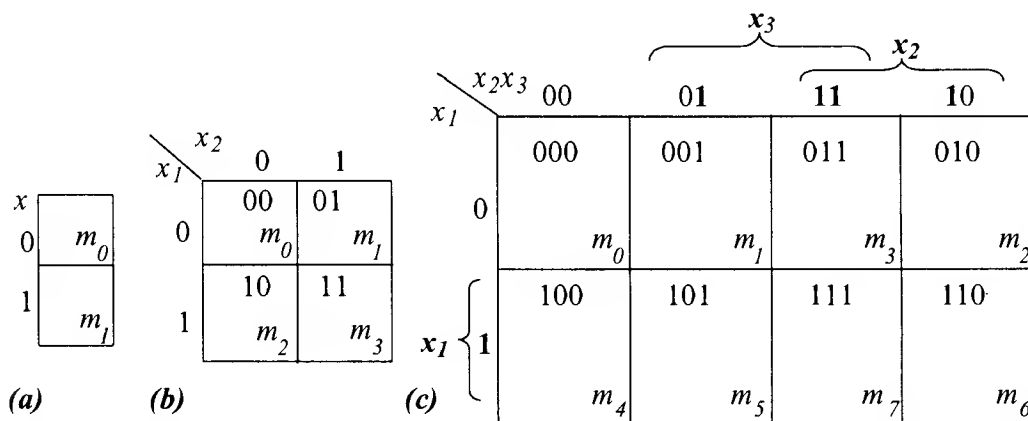
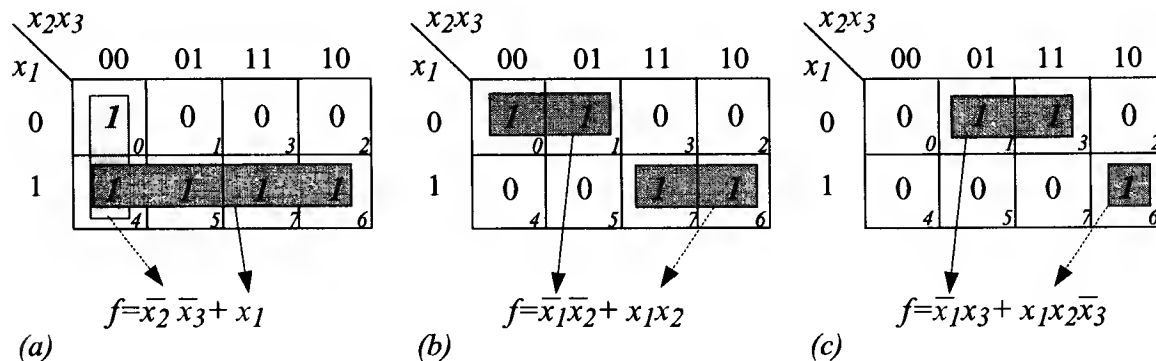


Figura 1.37. Simplificación de funciones usando diagramas de Karnaugh. Ver descripción en el texto.

La figura 1.38 muestra varios ejemplos. La presencia de "1" (existencia del minterm) en cuadros adyacentes nos indica la existencia de variables redundantes que agrupamos en múltiplos de  $2^n$ . Con  $n = 0$ ,  $2^0 = 1$ , no eliminamos ninguna variable. Con  $n = 1$ ,  $2^1 = 2$ , unimos dos celdas y eliminamos una variable. Con  $n = 2$ ,  $2^2 = 4$ , unimos cuatro celdas y eliminamos dos variables. Recuérdese además que una misma celda podemos usarla tantas veces como queramos ( $x + x = x$ ).



**Figura 1.38.** Ejemplos e minimización por diagramas de V-K.

En la figura 1.38.a, minimizamos la función  $f = \Sigma(m_0, m_4, m_5, m_6, m_7)$ . Como podemos observar agrupamos por una parte los elementos vecinos  $m_0$  y  $m_4$  obteniendo la expresión  $\bar{x}_2\bar{x}_3$  y por otra parte agrupamos  $m_4, m_5, m_6, m_7$ , que dan lugar a  $x_1$ . Así la función minimizada es:

$$f = \Sigma(m_0, m_4, m_5, m_6, m_7) = x_1 + \bar{x}_2\bar{x}_3 \quad [1.67]$$

No siempre se podrá realizar una simplificación tan fuerte. La figura 1.38.b muestra los ejemplos correspondiente a la función  $f = \Sigma(m_0, m_1, m_6, m_7)$ . En este caso agrupamos por una parte ( $m_0$  y  $m_1$ ) que da lugar a  $\bar{x}_1\bar{x}_2$  y por otra ( $m_6$  y  $m_7$ ) que proporciona  $x_1x_2$ . Quedando la función ya minimizada como:

$$f = \Sigma(m_0, m_1, m_6, m_7) = \bar{x}_1\bar{x}_2 + x_1x_2 \quad [1.68]$$

Podemos comprobar que el resultado es el mismo que el que se obtiene de la aplicación directa de los postulados y teoremas. Así,

$$m_0 + m_1 = \bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_1\bar{x}_2x_3 = \bar{x}_1\bar{x}_2 \quad [1.69]$$

$$m_7 + m_6 = x_1x_2x_3 + x_1x_2\bar{x}_3 = x_1x_2 \quad [1.70]$$

Análogamente, para el caso de la figura 1.38.c, la función a minimizar es  $f = \Sigma(m_1, m_3, m_6)$ . En este caso sólo podemos agrupar  $m_1$  y  $m_3$ . El otro término mínimo ( $m_6$ ) no se puede agrupar y queda como está. Por tanto la función resultante es:



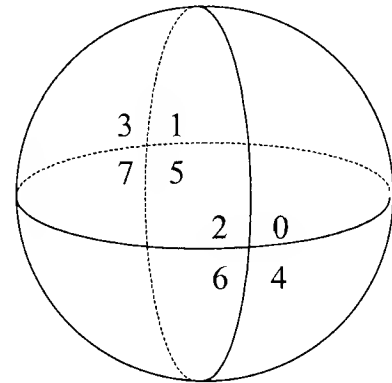
$$f = \bar{x}_1 x_3 + x_1 x_2 \bar{x}_3$$

[1.71]

Hay una serie de consejos parecidos a un procedimiento general de minimizar usando los mapas que dan buenos resultados en muchos casos. Son los siguientes:

1. *Agrupar las áreas que contengan "1" y que sean adyacentes.*
2. Recordar que son *áreas de forma cuadrada o rectangular*. Es decir han de ser 2, 4, 8, ...  $2^n$  el número de "1" adyacentes que se agrupan.
3. *El mapa se puede considerar como una esfera*. Es decir, el vecino del último por la derecha es el primero por la izquierda (de la misma fila), el vecino del último por debajo es el primero por arriba (de la misma columna). Así, las cuatro esquinas "están juntas" y por tanto pueden formar una misma área.

$\begin{array}{c} YZ \\ \swarrow \searrow \\ X \end{array}$	$\begin{array}{c} YZ \\ \Downarrow \\ 00 \end{array}$	$\begin{array}{c} YZ \\ \Downarrow \\ 01 \end{array}$	$\begin{array}{c} YZ \\ \Downarrow \\ 11 \end{array}$	$\begin{array}{c} YZ \\ \Downarrow \\ 10 \end{array}$
$X \Rightarrow 0$	0	1	3	2
$X \Rightarrow 1$	4	5	7	6



4. Una vez formados los agrupamientos *minimizarlos uno a uno usando adyacencia y absorción y sumar los resultados* obtenidos.
5. Recordar que *a veces es mejor usar los "0"* (términos que no están contenidos en la función) en vez de los "1" (términos que si los contiene). Esto se suele hacer cuando el número de "1" es pequeño mientras que el de "0" es grande. Si se minimiza agrupando ceros, recordar que *en este caso hay que complementar el resultado*.

Figura 1.39.

Veamos ahora algunos ejemplos.

**Ejercicio:** Minimizar por Karnaugh las siguientes funciones de tres variables:

a)  $f = \bar{X}\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + \bar{X}YZ + XYZ = \sum m(0,1,2,3,7)$

b)  $f = \bar{X}\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + \bar{X}YZ + X\bar{Y}\bar{Z} + XYZ = \sum m(0,2,3,4,7)$

c)  $f = \bar{X}\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + \bar{X}YZ + X\bar{Y}\bar{Z} + X\bar{Y}Z = \sum m(0,1,2,3,4,6)$

d)  $f = \bar{X}\bar{Y}\bar{Z} + \bar{X}YZ + X\bar{Y}Z + XYZ = \sum m(0,3,5,6)$

**Solución:**

a)  $f = \bar{X}\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + \bar{X}YZ + XYZ = \sum m(0,1,2,3,7)$

[1.72]

$\begin{matrix} YZ \\ \diagdown \\ X \end{matrix}$	$\bar{Y}\bar{Z}$ ↓ 00	$\bar{Y}Z$ ↓ 01	$YZ$ ↓ 11	$Y\bar{Z}$ ↓ 10
$\bar{X} \Rightarrow 0$	1 0	1 1	1 3	1 2
$X \Rightarrow 1$	0 4	0 5	1 7	0 6

$$f = \bar{X} + YZ$$

**Figura 1.40.** Minimización por diagramas de V-K de la función  $f = \sum m(0,1,2,3,7)$ .

En este caso, al ser el número de “0” menor que el de “1” podíamos haber minimizado la función  $f$  agrupando los “0” en vez de los “1”. Sin embargo el resultado es el mismo como veremos a continuación. Si agrupamos los “0”, quiere decir que vamos a trabajar con el complementario de la función  $f$ , por tanto lo único que tenemos que tener en cuenta es que la función que obtenemos es  $\bar{f}$ .

Así,

$YZ$	$\bar{Y}\bar{Z}$	$\bar{Y}Z$	$YZ$	$Y\bar{Z}$
$X$	$\Downarrow$ 00	$\Downarrow$ 01	$\Downarrow$ 11	$\Downarrow$ 10
$\bar{X} \Rightarrow 0$	1	1	1	1
$X \Rightarrow 1$	0	0	1	0

0 1 3 2  
4 5 7 6

$\bar{f} = X\bar{Z} + X\bar{Y}$

Figura 1.41. Minimización por diagramas de V-K de la función  $f = \Sigma m(4, 5, 6)$ .

Si ahora complementamos el resultado para obtener la función  $f$  y aplicamos los Teoremas de DeMorgan, vemos que la expresión que se obtiene es la misma que obtuvimos cuando minimizamos agrupando los “1”.

En efecto:  $f = \overline{X\bar{Z} + X\bar{Y}} = \overline{X(\bar{Z} + \bar{Y})} = \bar{X} + \overline{(\bar{Z} + \bar{Y})} = \bar{X} + YZ$  [1.73]

b)  $f = \bar{X}\bar{Y}\bar{Z} + \bar{X}Y\bar{Z} + \bar{X}YZ + X\bar{Y}\bar{Z} + XYZ = \Sigma m(0, 2, 3, 4, 7)$  [1.74]

$YZ$	$\bar{Y}\bar{Z}$	$\bar{Y}Z$	$YZ$	$Y\bar{Z}$
$X$	$\Downarrow$ 00	$\Downarrow$ 01	$\Downarrow$ 11	$\Downarrow$ 10
$\bar{X} \Rightarrow 0$	1	0	1	1
$X \Rightarrow 1$	1	0	1	0

0 1 3 2  
4 5 7 6

$f = \bar{Y}\bar{Z} + YZ + \bar{X}Y$

Figura 1.42. Minimización por diagramas de V-K de la función  $f = \Sigma m(0, 2, 3, 4, 7)$ .

c)  $f = \overline{X} \overline{Y} \overline{Z} + \overline{X} \overline{Y} Z + \overline{X} Y \overline{Z} + \overline{X} Y Z + X \overline{Y} \overline{Z} + X Y \overline{Z} = \sum m(0,1,2,3,4,6)$  [1.75]

En esta función, dado que el número de “0” es menor que el de “1” la minimizamos tomando los ceros. Así, obtenemos:

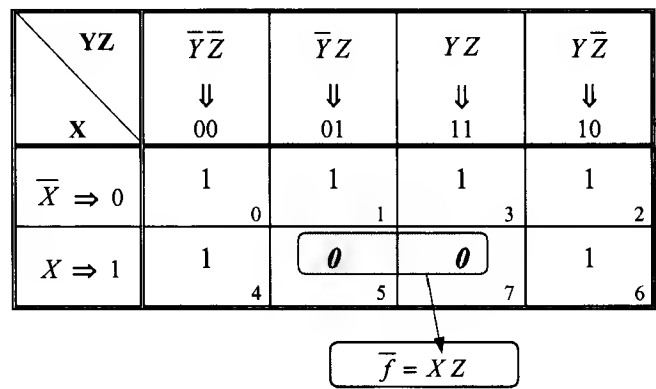


Figura 1.43. Minimización por diagramas de V-K de la función  $f=\sum m(0,1,2,3,4,6)$

Ahora para obtener la expresión de la función  $f$  minimizada deberemos complementar el resultado. Es decir,

$f = \overline{\overline{XZ}} = \overline{X} + \overline{Z}$  [1.76]

Es fácil comprobar que el resultado es el mismo que si hubiéramos operado sobre los “1”

d)  $f = \overline{X} \overline{Y} \overline{Z} + \overline{X} Y Z + X \overline{Y} Z + X Y \overline{Z} = \sum m(0,3,5,6)$  [1.77]

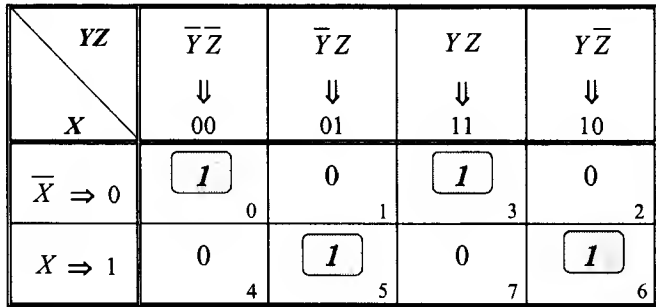


Figura 1.44. Diagrama V-K de la función  $f=\sum m(0, 3,5,6)$ .

En esta función vemos que no podemos agrupar sus términos para formar áreas mayores que cubran dos o más términos mínimos, esto es debido a que la función *ya es mínima* y no se puede simplificar más.

**Ejercicio:** Minimizar por Karnaugh las siguientes funciones de cuatro variables:

a)  $f = \sum m(0,2,3,4,5,6,7,8,12)$

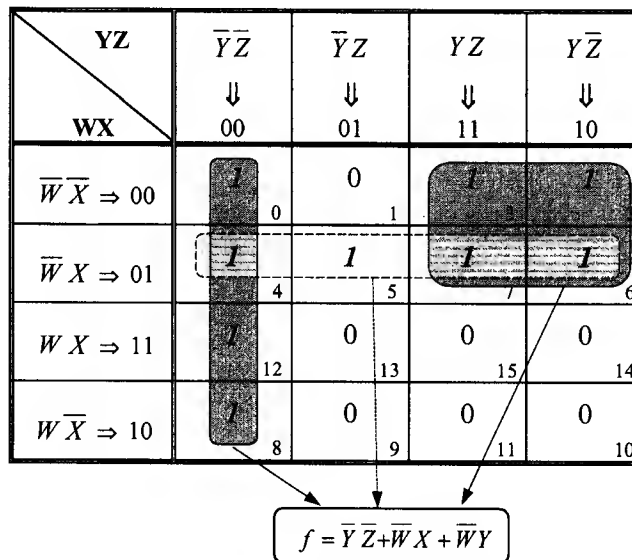
b)  $f = \sum m(0,2,4,8,10,12)$

c)  $f = \sum m(1,3,6,9,13,14,15) + \sum d(8,11,12)$

**Solución:**

a)  $f = \sum m(0,2,3,4,5,6,7,8,12)$

[1.78]



**Figura 1.45.** Minimización mediante diagrama de V-K de la función de cuatro variables  $f = \sum m(0,2,3,4,5,6,7,8)$ .

b)  $f = \sum m(0,2,4,8,10,12)$

[1.79]

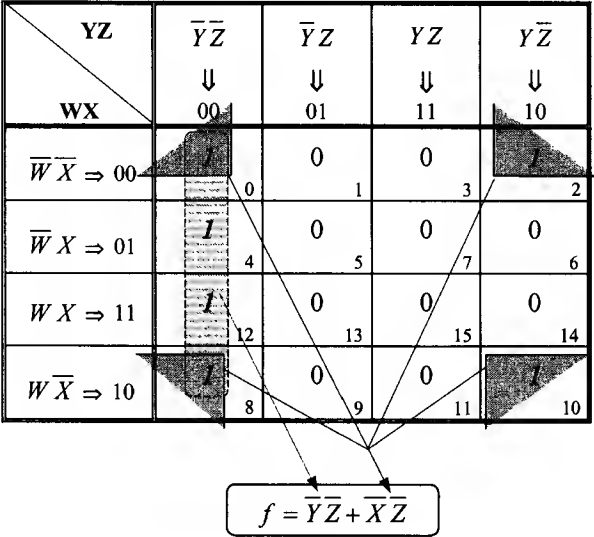


Figura 1.46 Minimización mediante diagrama de V-K de la función  $f = \Sigma m(0, 2, 4, 8, 10, 12)$ .

En este caso es más fácil si se resuelve agrupando los ceros, como podemos observar en la figura 1.47. El motivo de haberlo resuelto agrupando los “1” ha sido para poner de manifiesto el hecho de que las cuatro esquinas son también términos mínimos vecinos y por tanto los podemos agrupar.

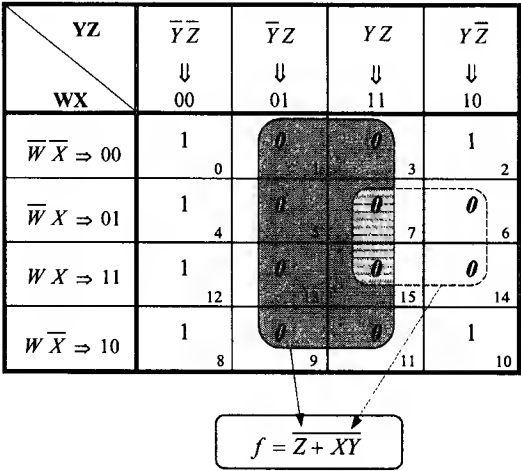


Figura 1.47. Minimización de la función  $f = \Sigma m(0, 2, 4, 8, 10, 12)$  agrupando los ceros.

$$c) \quad f = \sum m(1,3,6,9,13,14,15) + \sum d(8,11,12) \quad [1.89]$$

Los términos mínimos 8, 11 y 12 son términos que son indiferentes, es decir que no son importantes para la función y que por tanto pueden tomar el valor 0 ó el valor 1. Estos términos mínimos los señalamos en la tabla como  $d$ , y los agruparemos con los “1” (o con los “0”) siempre que nos ayuden a obtener áreas mayores, pero entendiendo que ningún área podrá estar formada sólo por términos de este tipo. Así,

YZ WX	$\bar{Y}\bar{Z}$ ↓ 00	$\bar{Y}Z$ ↓ 01	$YZ$ ↓ 11	$Y\bar{Z}$ ↓ 10
$\bar{W}\bar{X} \Rightarrow 00$	0 0	1 1	1 3	0 2
$\bar{W}X \Rightarrow 01$	0 4	0 5	0 7	1 6
$WX \Rightarrow 11$	$d$ 12	1 13	1 15	1 14
$W\bar{X} \Rightarrow 10$	$d$ 8	1 9	$d$ 11	0 10

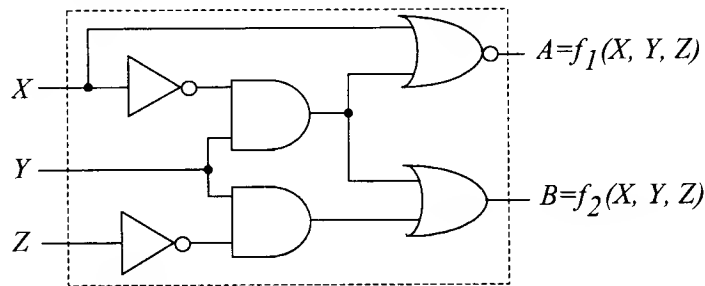
$$f = WX + \bar{X}Z + XY\bar{Z}$$

**Figura 1.48.** Minimización de la función  $f = \sum m(1,3,6,9,13,14,15) + \sum d(8,11,12)$ .

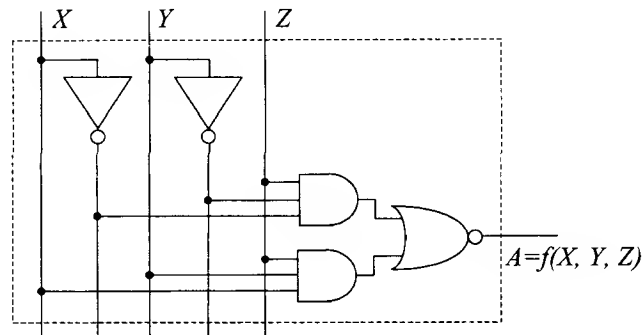
No queremos terminar este apartado sobre minimización sin hacer algunos comentarios. Primero, es conveniente recordar que aunque el problema de la minimización se ha presentado aquí de forma aparentemente sencilla, cuando aumenta el número de variables la complejidad es tal que sólo es abordable con la ayuda del computador y con un software específico que hace computacional un conjunto de procedimientos complejos. Por otro lado, la evolución de la tecnología integrada hace que el problema de la minimización se plantee en otros términos tales como número máximo de terminales en un circuito, consumo, velocidad de conmutación, carácter repetitivo del diseño y posibilidad de soluciones programadas.

## 1.8. PROBLEMAS (\*)

**E.1.1.** Obtener las expresiones lógicas que describen las relaciones entre las variables de entrada  $X$ ,  $Y$  y  $Z$  y las variables de salida  $A$  y  $B$  en el circuito de la figura.



**E.1.2.** Analizar el circuito de la figura y pasarlo a su equivalente en puertas NAND.



**E.1.3.** Pasar a puertas NOR el circuito del ejercicio anterior.

**E.1.4.** Sintetizar a) usando AND, OR y Negación, b) sólo puertas NAND, c) sólo puertas NOR, el circuito correspondiente a las siguientes funciones lógicas:

$$A = f_1(X, Y, Z) = (\bar{X}Y + X\bar{Y})Z + \bar{Z}(\bar{X}Y + \bar{X}\bar{Y})$$

$$B = f_2(X, Y, Z) = \bar{X} + \bar{Y}Z + \bar{Z}X$$

(\*) La solución de estos ejercicios se encuentra en el texto de problemas.



- E.1.5.** Representar en forma canónica de términos mínimos las funciones  $f_1$  y  $f_2$  del ejercicio anterior. Cambiar la representación a maxterms y explicar los aspectos puntuales del principio de dualidad sobre ambos ejemplos.
- E.1.6.** Minimizar por aplicación reiterativa de los teoremas de DeMorgan y del teorema de adyacencia las siguientes funciones:

$$A = f_1(X, Y, Z) = X\bar{Y} + Y\bar{Z} + \bar{X}\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + XYZ$$

$$B = f_2(X, Y, Z) = (X + Y + \bar{Z}) + (X + \bar{Y} + \bar{Z}) + \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z}$$

*Demostrar por inducción completa y por diagramas que las expresiones simplificadas son realmente equivalentes a las expresiones iniciales.*

- E.1.7** Usar los diagramas de Karnaugh para simplificar las siguientes funciones

$$A = f_1(X, Y) = \bar{X}\bar{Y} + X\bar{Y} + \bar{X}Y$$

$$B = f_2(X, Y, Z) = \bar{X}\bar{Y}Z + (\bar{Y} + \bar{Z})X + Y$$



## ***Preparación de la Evaluación***

---

*A lo largo del tema hemos ido incluyendo distintos ejercicios para ayudar al lector a comprender los conceptos expuestos y, a la vez, para ayudarle a preparar sus pruebas de evaluación de forma que los contenidos de estas pruebas no van a ser, en general, muy diferentes de los ejercicios que aquí se presentan.*

*Por otro lado hemos incluido una relación de problemas procedentes de la anterior edición de este texto cuyas soluciones se encuentran en un texto complementario.*

*Aún a costa de ser redundantes, terminamos el tema con este apartado de conocimiento tutorial que incluye nuevos ejercicios análogos a los resueltos previamente junto con algunas reflexiones de carácter conceptual resultado de la experiencia docente y que pretenden “adivinar” y ayudar a resolver las dudas potenciales del lector. Estamos profundamente convencidos de que en la enseñanza a distancia la evaluación del alumno empieza con la preparación del material didáctico, porque en la estructura de ese material debe estar subyacente la prueba de evaluación, dejando claro qué es lo que queremos que sepa el alumno.*

*Recordemos que hemos descompuesto el contenido de este tema en 6 objetivos. Por consiguiente, la ayuda a la evaluación también la vamos a descomponer de acuerdo con esos objetivos.*

Objetivos
<ul style="list-style-type: none"><li>1: Analógico versus Digital</li><li>2: Postulados y Teoremas</li><li>3: Representación</li><li>4: Análisis</li><li>5: Síntesis</li><li>6: Minimización</li></ul>



*Para cada uno de estos objetivos se proponen una o más cuestiones teóricas y algunos*

*ejercicios cuyas soluciones se pueden obtener por analogía con las soluciones a otros problemas análogos intercalados en la teoría o en el texto base de problemas.*

## **1. Analógico versus Digital**

- 1.1. *¿Entiende la distinción entre dato y operador?. Úsela para describir las diferencias entre cálculo analógico y cálculo digital.*
- 1.2. *¿Cuántos valores puede tener una señal analógica?. ¿Y una digital?. ¿Cómo se puede hacer más precisa la representación digital de una magnitud física continua?*
- 1.3. *“El tiempo analógico es continuo y el digital es discreto”. ¿Podría explicar el significado de esta afirmación?.*
- 1.4. *¿Ha comprendido la distinción entre reglas de transformación propias del cálculo analógico y reglas propias del cálculo digital?. ¿Podría enumerar un conjunto de operadores analógicos y otro de operadores digitales (binarios)?.*

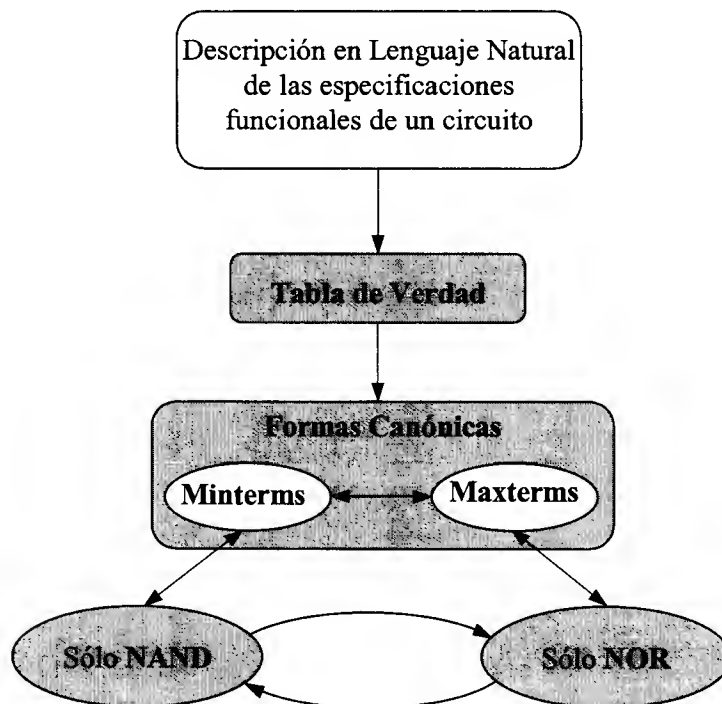
## **2. Postulados y Teoremas**

- 2.1. *¿Qué significa que “el producto es distributivo respecto de la suma?. ¿Cómo podría demostrarlo?. Hemos incluido en el tema la demostración de que  $x(y+z) = xy+xz$ . ¿Podría demostrar ahora la expresión dual, es decir, que también la suma es distributiva respecto del producto?.*
- 2.2. *Hemos visto la demostración de los teoremas de Absorción ( $x+xy=x$ ) y Adyacencia ( $x\bar{y} + x\bar{y} = x$ ). Demuestre ahora las expresiones duales ( $x(x+y)=x$ ) y  $(x+y)(\bar{x}+\bar{y})=x$*
- 2.3. *Use un simulador tipo PSpice o equivalente para comprobar todos los teoremas. Recuerde que un procedimiento para generar de forma automática las cuatro configuraciones de valores posibles para dos variables lógicas (00, 01, 10, 11) es usar como señales de entrada dos relojes en los que la frecuencia de uno de ellos es el doble de la del otro.*

## **3. Representación**

*Sabemos que toda función lógica puede representarse mediante tablas de verdad o por expresiones lógicas en términos de algún conjunto completo de operadores (AND, OR, NOT; sólo*

*NAND, sólo NOR). Sabemos, además que usando el conjunto completo de operadores (AND, OR, NOT), existen dos formas canónicas duales para representar una misma función lógica: suma de productos y producto de sumas. Por consiguiente, pueden plantearse distintas pruebas de evaluación en este apartado que tiene que ver con los distintos tipos de representación y con los cambios de uno a otro, tal como se ilustra en la siguiente figura.*



*Así pues, para preparar la evaluación de este objetivo, conviene recordar que los pasos a seguir son los siguientes:*

1. **Escribir una función arbitraria con pocas variables para que el problema sea manejable**  $\Rightarrow A = f_1(X, Y, Z)$
2. **Seleccionar una forma de representación** (sólo NAND, por ejemplo) y obtener la versión de esa función usando sólo este tipo de operador.
3. **Cambiar la representación** (a sólo NOR, por ejemplo)
4. **Comprobar con la tabla de verdad y/o con el simulador** que ambas representaciones (2) y (3) coinciden con (1).

3.1. Compruebe que ha comprendido el procedimiento pasando a sólo NAND las siguientes funciones:

$$A = \overline{X}YZ + X\overline{Y}Z + \overline{X}Y\overline{Z}$$

$$A = \sum m(1,3,5,7)$$

$$A = \prod M(1,3,5)$$

3.2. Dibuje un circuito que sólo tenga puertas NOR y sintetice las siguientes funciones lógicas:

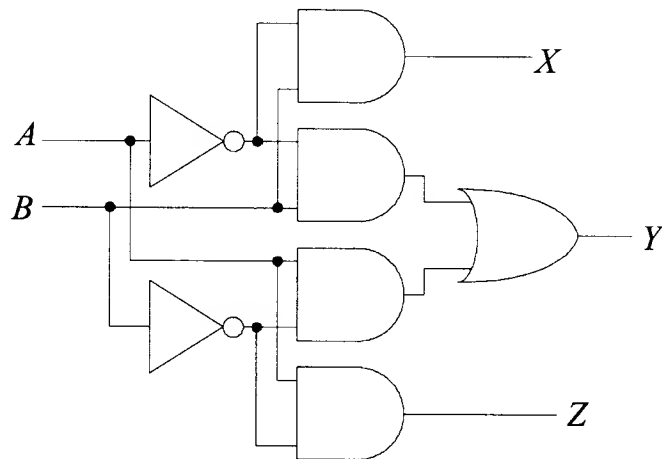
$$A = \overline{Y}Z + YZ + \overline{X}Y$$

$$A = \overline{X}Y\overline{Z} + \overline{X}YZ + X\overline{Y}Z + XY\overline{Z}$$

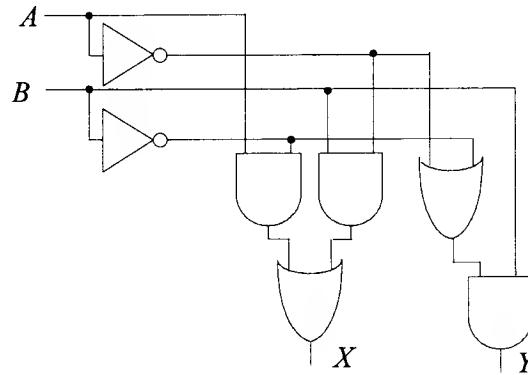
#### 4. Análisis

Recuerde que analizar un circuito lógico es encontrar la función lógica que calcula, a partir del esquema de conexión de las variables de entrada con los distintos operadores. Compruebe que ha comprendido el concepto y los ejemplos incluidos en la teoría analizando los siguientes circuitos:

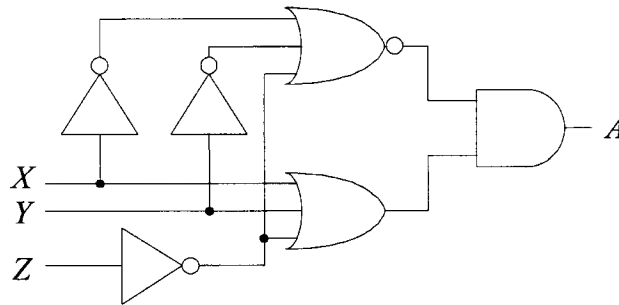
4.1.



4.2.



4.3.



## 5. Síntesis

*Recuerde que sintetizar un circuito es:*

- a) *Pasar de una descripción en lenguaje natural a una función lógica.*
- b) *Pasar de una función lógica a su circuito correspondiente.*

**5.1.** *Necesitamos dos circuitos, uno que pase las representaciones en binario puro de los ocho primeros números (000, 001, ..., 111) a los decimales correspondientes (0, 1, ..., 7) y otro circuito que realice el cambio de código inverso. Es decir, que pase de ocho líneas (0, 1, ..., 7) a tres líneas de salida que codifiquen en binario estos dígitos. ¿Sabrías diseñar estos circuitos?*

**5.2.** *Sintetizar e implementar un circuito con tres entradas y una salida, de forma que esta última se ponga en “alta” siempre que a la entrada exista un número de ceros mayor o igual que dos.*

**5.3.** *Cuatro amigos tienen que decidir si ir al cine, ir de copas o quedarse estudiando y quieren tomar la decisión por votación. La votación consiste en que cada uno tiene una moneda y en el mismo instante los cuatro amigos deben presentar a la vez la moneda, de forma que si el número de caras es mayor que el de cruces irán al cine, si es menor van de copas y si hay empate se quedan estudiando. Sintetizar, minimizar e implementar un circuito que realice esta función.*

**5.4.** *Sintetizar la función:*

$$Y = A\bar{B}C + A\bar{B}\bar{C} + \bar{A}BC + \bar{A}\bar{B}$$

## 6. Minimización

*Recordemos que lo que debe autoevaluarse el alumno en este apartado es:*

- a.** *Cierta habilidad para **minimizar funciones lógicas por manipulación algebraica y uso directo de los postulados y teoremas del Álgebra de Boole**. El ejercicio E.1.6 del texto de problemas es representativo de esta forma de minimizar y en el material teórico de este tema se han incluido varios ejercicios resueltos de este tipo.*
- b.** *Su grado de comprensión de los Mapas de Karnaugh que reordenan los términos mínimos para facilitar la minimización por simple inspección visual de los agrupamientos más convenientes.*

- 6.1.** *¿Qué teoremas son los más adecuados para la minimización algebraica?. ¿Por qué?. ¿Qué teorema se usa para pasar de  $\bar{X}YZ + \bar{X}Y\bar{Z}$  a  $\bar{X}Y$ ?. ¿Podría describirlo con palabras?*
- 6.2.** *¿Sabría usted construir el Mapas de Karnaugh para dos variables?. ¿Podría minimizar las funciones  $f_1 = \sum m(0,1,2)$  y  $f_2 = \prod M(1,2,3)$ ?*
- 6.3.** *¿Sabría construir el Mapa de Karnaugh para funciones de tres variables?. ¿Por qué se cambia el orden de las columnas pasando del “natural” (00,01,10,11) al de ahora (00,01,11,10)?*
- 6.4.** *Minimizar las siguientes funciones de tres variables. Obsérvese que son las complementarias de algunos de los ejercicios resueltos en el tema. ¿En qué le ayuda al proceso de minimización el conocer el carácter complementario de estas funciones?*

a)  $f_1 = \sum m(1,5,6)$

b)  $f_2 = \sum m(5,7)$

c)  $f_3 = \sum m(1,2,4,7)$

6.5. Minimizar las siguientes funciones de cuatro variables.

a)  $f_1 = \sum m(1,9,10,11,13,14,15)$

b)  $f_2 = \sum m(1,3,5,6,7,9,11,13,14,15)$



## REFERENCIAS BIBLIOGRÁFICAS

1. Delgado, A.E., Mira, J., Hernández R., Lázaro. J.C.: “Problemas de Electrónica Digital”. Sanz y Torres. Madrid, 1995, 1999.
2. Fletcher, W.I.: “*Engineering Approach to Digital Design*”. Prentice Hall, 1980.
3. Sandige, R.S.: “*Modern Digital Design*”. McGraw-Hill. 1990.





# TEMA 2

---

## ***Dispositivos Semiconductores en Corte y Saturación***

---

### **✦ CONTEXTO**

*El propósito general de este tema es resumir los conocimientos de Electrónica Física y Física de Dispositivos Electrónicos sobre el comportamiento de diodos y transistores.*

*En el tema 1 hemos visto los modelos lógicos que subyacen a los circuitos combinacionales y el conjunto de operadores (puertas AND, OR, NOT; NAND y NOR) necesarios y suficientes para sintetizar cualquier circuito lógico.*

*Por otro lado, en los temas 3 y 4 estudiaremos la forma de construir físicamente esos operadores usando transistores bipolares y de efecto campo. Por eso es necesario este capítulo 2º que nos sirve de puente entre la Física de Dispositivos (comprender el funcionamiento de los*

transistores) y la síntesis de los operadores lógicos (usar esos transistores para diseñar circuitos NAND o NOR).

Más adelante, en los temas sobre lógica programable (tema 7) y memorias RAM (tema 11) volveremos a hacer uso de estos conocimientos de Física de Dispositivos.

## ++ CONOCIMIENTO PREVIO NECESARIO

Son convenientes ciertos conocimientos previos sobre Electrónica Física y Física de Dispositivos porque este tema es bastante compacto. Estos conocimientos se suelen encontrar en asignaturas de “Fundamentos Físicos de la Informática”, cursadas usualmente al comienzo del primer curso.

Afortunadamente, para el estudio de la Electrónica Digital (salvo en puertas ECL), los alumnos sólo necesitan conocer el comportamiento en corte y saturación de los diodos y transistores que actúan como conmutadores. Por eso creemos que es posible resumir en un tema estos conocimientos.

## +++ OBJETIVOS DEL TEMA

**Objetivo 1:** *Comprender el funcionamiento de los diodos de unión PN considerados como elementos de circuito y en situaciones de corte y saturación (como conmutador).*

- *Saber cómo se polariza un diodo y cómo cambian los valores de corriente y tensión en sus extremos al variar los valores de la tensión de entrada.*
- *Saber utilizar un simulador para obtener las curvas características del diodo y visualizar su comportamiento como rectificador.*

**Objetivo 2:** *Comprender el principio de funcionamiento de los transistores bipolares.*

- *Comprender el significado de las características externas y de transferencia.*
- *Saber polarizar un transistor en directa y en inversa.*
- *Conocer el circuito equivalente para pequeñas y grandes señales y tener muy claro los valores de tensión y corriente característicos de los estados de corte y saturación que*

*vamos a usar de forma repetida en el tema 3 para explicar cómo funcionan las puertas lógicas.*

**Objetivo 3:** *Comprender el funcionamiento de los transistores MOS, que van a dar origen a las familias de puertas lógicas MOS y CMOS, al igual que los transistores bipolares dan origen a toda la tecnología bipolar y a las familias TTL y ECL.*

- *Este objetivo se desglosa en los mismos apartados que el anterior: comprender el funcionamiento físico, conocer sus curvas características, saber polarizar al dispositivo y conocer sus modelos para pequeñas señales y en condiciones de conmutación.*

### ++++ GUÍA DE ESTUDIO

*La guía de estudio para este tema es muy dependiente del perfil del alumno. El consejo para el alumno estándar es que siga el contenido de este tema y el correspondiente del texto de problemas teniendo siempre un simulador muy cerca. Este simulador es algo parecido al “alumno ideal”, que se lo sabe todo, de forma que si las formas de onda que nosotros creemos que tienen que aparecer, no aparecen al simular, algo no ha sido hecho correctamente.*



---

## ***Contenido del Tema***

---

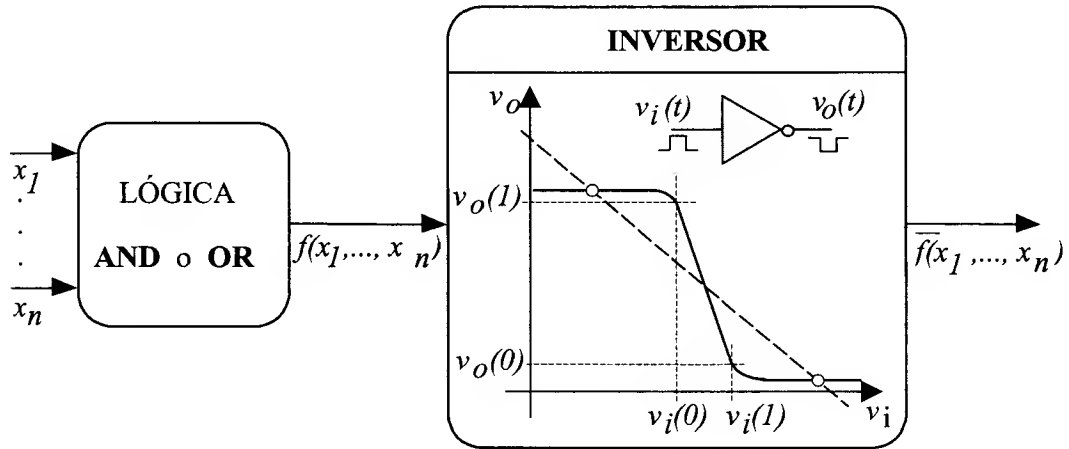
### **2.1. ESTRUCTURA GENERAL DE LOS OPERADORES LÓGICOS**

En el tema anterior hemos visto el modelo formal subyacente a toda la computación digital. Para la síntesis de cualquier circuito combinacional nos basta con un conjunto completo de operadores lógicos. En este tema vamos a ver las bases de electrónica física y de física de dispositivos electrónicos necesarias para la síntesis electrónica de esos operadores lógicos que tienen una estructura general análoga a la que se ilustra en la *figura 2.1*. Todos ellos poseen una parte inicial que se encarga de realizar la lógica AND ó OR mediante diodos, uniones PN en transistores bipolares o tecnología MOS. Después la síntesis del operador se completa con una segunda parte encargada de sintetizar la negación. Es decir, mediante un circuito inversor. Veremos primero las bases electrónicas necesarias para la síntesis de los inversores y, más adelante, al estudiar las distintas familias lógicas, veremos cómo se sintetiza la parte inicial encargada de realizar las funciones AND y OR.

Un inversor es un circuito con dos estados,  $v(1)=V_H$  (tensión del estado de alta) y  $v(0)=V_L$ , (tensión del estado de baja) y una característica de transferencia como la que se ilustra en la *figura 2.1*. Mientras la tensión de entrada  $v_i(t)$  es inferior a un cierto valor umbral,  $v_i(0)$ , la salida está en alta, a un valor constante,  $v_o(1)$ . Si la tensión de entrada sigue aumentando, el circuito conmuta, para  $v_i(t)=v_i(1)$ , a un valor bajo constante,  $v_o(0)$  y allí permanece para posteriores incrementos de  $v_i(t)$ .

Como en realidad la asignación de un estado lógico a un nivel de tensión en un circuito es de naturaleza arbitraria, hay dos convenios posibles: (1) *lógica positiva* y (2) *lógica negativa*. En el primer caso asociamos el "1" lógico al valor de tensión más alto. Es decir, en lógica positiva  $V(1)=V_H$  siempre es mayor que  $V(0)=V_L$ . Por ejemplo, 5 y 0 voltios. Inversamente, si asociamos el valor más alto de tensión al cero lógico, tendríamos lógica negativa. En adelante y salvo que se especifique lo contrario, usaremos siempre el criterio de lógica positiva.

En el caso ideal,  $v_o(1)$  y  $v_o(0)$  son niveles constantes y repetibles de un inversor a otro y la transición,  $v_i(0) \Rightarrow v_i(1)$ , es instantánea. Realmente las cosas no son así. Las transiciones *alta*  $\Rightarrow$  *baja* y *baja*  $\Rightarrow$  *alta* duran un cierto tiempo cuyo inverso es una medida de la velocidad de conmutación y los niveles de alta y baja poseen una banda de tolerancia debida esencialmente a la dispersión propia del proceso de fabricación de circuitos integrados, como veremos más adelante.



**Figura 2.1.** Estructura general de los operadores lógicos. Constan de una parte inicial encargada de la lógica AND o OR seguida de un circuito inversor que sintetiza la negación.

Hay muchas formas diferentes de sintetizar un circuito inversor, cada una de las cuales da lugar a una “*familia lógica*” caracterizada por un conjunto de parámetros estáticos y dinámicos de velocidad, inmunidad al ruido, consumo, niveles lógicos, etc. Las tecnologías de integración pueden ser *bipolar* o *unipolar* (MOS y C-MOS). A su vez, dentro de la tecnología bipolar existen soluciones en las que los dispositivos pasan de corte a saturación (por ejemplo en la familia TTL) y otras en las que los dispositivos no se saturan, trabajando siempre en zona activa (familia ECL).

Nuestra misión ahora es estudiar el operador básico en cada una de estas familias lógicas, pero la forma de hacerlo depende drásticamente del perfil del alumno. En efecto, el estudio de las distintas familias lógicas depende del nivel de conocimientos electrónicos previos en el alumno al que se supone dirigido este texto. Hay dos situaciones claras y distintas. Por un lado está el alumno que ha estudiado previamente asignaturas de electrónica física y física de dispositivos y que, por consiguiente conoce de forma razonable el funcionamiento de diodos y transistores bipolares y MOS tanto en

comportamiento lineal como en situaciones de corte y saturación, propias de la electrónica digital. Para este perfil de alumno no haría falta este tema, pudiendo pasar a la síntesis de los operadores básicos (NAND, por ejemplo) en las distintas familias y tecnologías.

Por otro lado está el alumno que se enfrenta a la electrónica digital sin estudios previos sobre física de semiconductores y dispositivos activos. Para este alumno está dirigido este tema que pretende mantener una solución de compromiso entre “*no decir nada*” de electrónica física y física de dispositivos y seguir trabajando sólo con los símbolos de los operadores lógicos, sin preocuparse de su síntesis y “*decir todo*” lo que debería de haber estudiado en otras asignaturas previas. El nivel de compromiso se ha establecido en la descripción de los dispositivos como *elementos de circuito* por sus características externas y de transferencia y por las ecuaciones mínimas necesarias para caracterizar su comportamiento dinámico en los transitorios del paso de corte a saturación y viceversa.

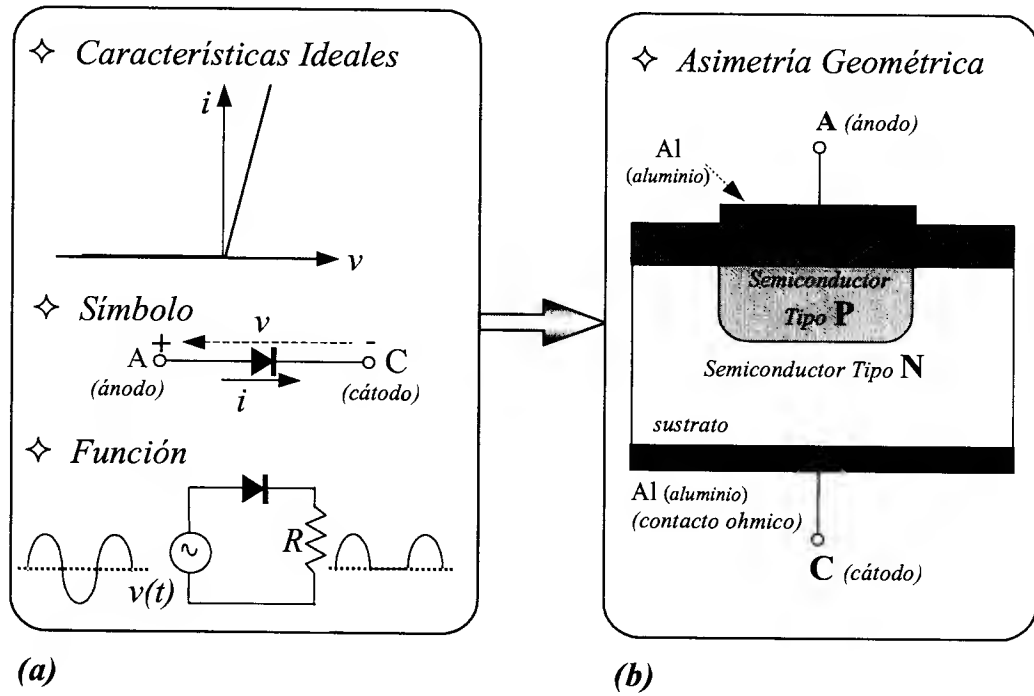
Veremos en este tema los *diodos de unión P-N*, los *transistores bipolares* y los *transistores MOS*. Para aquellos alumnos que no hayan tenido ningún contacto previo con estos contenidos se aconseja la lectura de algún libro sobre *física de semiconductores* que introduzca de forma mínima los conceptos necesarios para comprender el funcionamiento de los dispositivos semiconductores.<sup>(\*)</sup>

## 2.2. DIODOS DE UNIÓN P-N

Un diodo de unión P-N es un dispositivo electrónico que conduce prácticamente sin resistencia para tensiones positivas y no conduce para tensiones negativas, tal como se ilustra en la figura 2.2.a, donde aparece su símbolo, sus características tensión-corriente ( $v,i$ ) y la ilustración de su función como rectificador que muestra el carácter unidireccional de la conducción. Ante una onda que pasa alternativamente por semiciclos positivos y negativos, sólo deja pasar a la resistencia de carga,  $R$ , los semiciclos positivos porque en este caso su resistencia es muy baja. En cambio, elimina los semiciclos negativos porque su resistencia en este caso es muy alta.

---

(\*) Una referencia adecuada para complementar el contenido de todo este capítulo son los textos “*Electrónica Analógica Lineal*” de J. Mira y A. Delgado, editado por la UNED (ref. 074076) y “*Problemas Resueltos de Física de Dispositivos Electrónicos*” de E. Carmona y J. Mira, editado también por la UNED (ref. 07407).



**Figura 2.2.** Diodos semiconductores de unión P-N. (a) Características externas del caso ideal, símbolo y función rectificadora básica. (b) Estructura física.

Para sintetizar este dispositivo, la física de semiconductores nos dice que debemos introducir una discontinuidad en un material semiconductor pasando de un tipo P (exceso de huecos, positivos) a un tipo N (exceso de electrones, negativos) en un corto espacio. Aparece así un potencial de difusión o de contacto consecuencia del paso de la zona P a la zona N de cargas positivas móviles que dejan cargas iónicas inmóviles y del paso complementario de electrones de N a P, dejando cargas iónicas positivas. Tenemos así una barrera de potencial,  $v_d$ , que es la responsable de la conductividad asimétrica (el dispositivo conduce en un sentido pero no en el opuesto). Obsérvese que esta asimetría eléctrica es función de una asimetría geométrica en el semiconductor. Al existir una barrera de potencial,  $v_d$ , la tensión externa puede actuar aumentando esa barrera cuando se suma ( $v_d + v_a$ ) y no conduciendo o disminuyendo la barrera ( $v_d - v_a$ ), hasta prácticamente anularla, dando lugar a un proceso de conducción con baja resistencia.

Para conseguir esa discontinuidad se produce una difusión selectiva que hace que sobre un sustrato tipo N, por ejemplo, aparezca una región de tipo P. Esto se consigue haciendo crecer una capa

de óxido sobre la región de tipo N y abriendo después una ventana por procedimientos fotolitográficos que eliminan el óxido en una región concreta donde posteriormente se realizará la difusión que convierte en P a esa región del semiconductor superponiendo una densidad de impurezas aceptoras (tipo Boro, por ejemplo) que neutralizan las impurezas dadoras existentes en esa región. Posteriormente se establecen unos contactos metálicos, sobre la zona P y sobre el sustrato tipo N, que dan lugar a los terminales del dispositivo (*figura 2.2.b*).

Supongamos que ya tenemos construida la estructura de una unión P-N. El siguiente paso es caracterizar su comportamiento eléctrico como elemento de circuito, es decir calcular:

- a) Su *comportamiento estático*: curvas ( $v, i$ ) y polarización.
- b) Su *comportamiento dinámico*:
  - b.1) Para *pequeñas señales*.
  - b.2) En *corte y saturación*.

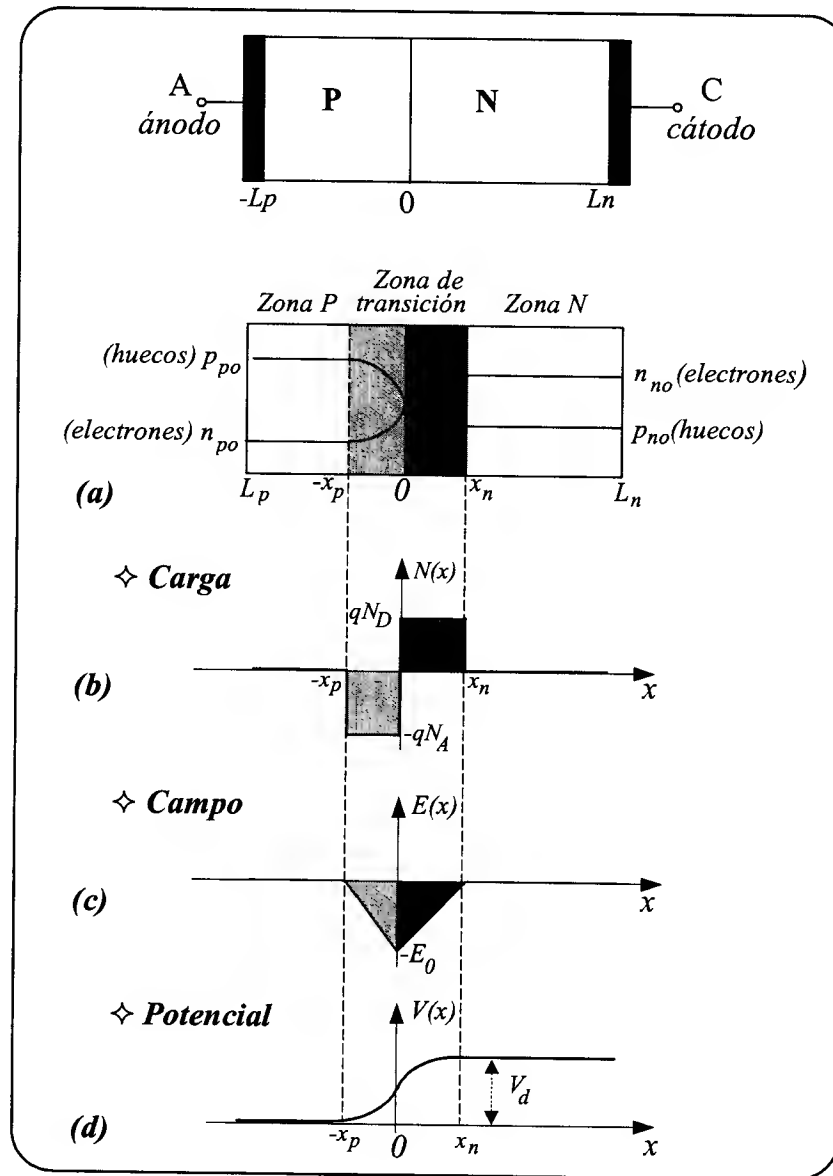
La *figura 2.3* resume el comportamiento estático. Se suele usar un modelo unidimensional y las hipótesis de impurificación uniforme y capa vacía de portadores móviles. De forma que en condiciones estacionarias y sin aplicar ninguna tensión externa ( $v_d=0$ ), aparecen tres zonas: *zona neutra P*, *zona de transición* y *zona neutra N* (*figura 2.3.a*). Al establecerse la unión aparecen *corrientes de difusión* debidas a la existencia de *gradientes de concentración* tanto para los huecos (hay más en la zona P,  $p_{po}$ , que en la N,  $p_{no}$ ) como para los electrones (hay más en la N,  $n_{no}$ , que en la P,  $n_{po}$ ). Al llegar los portadores (electrones o huecos) a la región donde son minoritarios se recombinan y dejan cargas iónicas inmóviles (positivas en N,  $qN_Dx_n$ , y negativas en P,  $qN_Ax_p$ , *figura 2.3.b*) que crean un campo que se opone a la difusión (*figura 2.3.c*). La situación de equilibrio corresponde al balance de las tendencias difusoras con las de arrastre, dando lugar a una *corriente total nula*.

Como resultado de este equilibrio aparece una *barrera de potencial*,  $v_d$  (*figura 2.3.d*) que, tal como comentamos anteriormente, es la causante de la conducción asimétrica del diodo. Veamos primero el cálculo de la barrera y después su modificación (polarización) mediante la aplicación de potenciales externos ( $\pm v_a$ ) que suben o bajan esa barrera ( $v_d \pm v_a$ ).

En la zona N hay una carga  $q \cdot N_D$  y en la zona P hay otra carga de signo opuesto  $-q \cdot N_A$ . Siendo  $N_D$  el número de átomos dadores y  $N_A$  el número de átomos aceptores que a la temperatura de trabajo



están completamente ionizados y, por consiguiente, contribuye cada uno con una carga iónica.



**Figura 2.3.** Modelo lineal usado para analizar el comportamiento estático de la unión P-N. **(a)** Distinción entre las tres zonas (P, N y de transición). **(b)** Perfil de la carga iónica inmóvil que va a dar lugar a la barrera de potencial. **(c)** Expresión del campo eléctrico resultado de integrar la carga. **(d)** Forma de la barrera de potencial, resultado de integrar el campo eléctrico.

Por simple aplicación de la ecuación de Poisson, obtenemos:

$$\text{Zona P: } \frac{dE}{dx} = -\frac{qN_A}{\varepsilon} \quad \text{para } x_p \leq x \leq 0 \quad [2.1]$$

luego la expresión del campo en esa región será la ecuación de una recta

$$E = -\frac{qN_A}{\varepsilon}(x + x_p) \quad [2.2]$$

Análogamente para la zona N tendremos:

$$\frac{dE}{dx} = -\frac{qN_D}{\varepsilon} \quad \text{para } 0 \leq x \leq x_n \quad [2.3]$$

luego el valor del campo en esta región será:

$$E = \frac{qN_D}{\varepsilon}(x - x_n) \quad [2.4]$$

Por consiguiente, la altura de la barrera de potencial,  $V_d$ , se obtiene integrando el campo. Es el área bajo el campo con signo cambiado,

$$V_d = \int_{-x_p}^{x_n} -E(x) \cdot dx = \frac{x_n + x_p}{2} \cdot E_0 = q \frac{N_D \cdot x_n}{2\varepsilon} (x_n + x_p) \quad [2.5]$$

$$\text{o bien, } V_d = \frac{q \cdot N_A}{2\varepsilon} x_p (x_n + x_p) \quad [2.6]$$

$$\text{ya que, por neutralidad eléctrica, } N_D \cdot x_n = N_A \cdot x_p \quad [2.7]$$

Si llamamos  $W$  al ancho total de la zona de transición, ( $|x_n| + |x_p|$ ), obtenido sumando las penetraciones en las zonas N, ( $x_n$ ), y P, ( $x_p$ ), y añadiendo al potencial de difusión  $V_d$ , el que se aplica externamente al polarizar el dispositivo en sentido directo para disminuir la barrera ( $-V_a$ ) o en sentido inverso para aumentarla ( $+V_a$ ), tenemos:

$$W = |x_n| + |x_p| = \sqrt{\frac{2(V_d \pm V_a)}{q \cdot N_B}} \quad \text{siendo } N_B = \frac{N_A \cdot N_D}{N_A + N_D} \quad [2.8]$$

## 2.3. COMPORTAMIENTO ESTÁTICO, POLARIZACIÓN Y COMPORTAMIENTO DINÁMICO.

Veamos ahora las *características tensión-corriente*,  $I=f(V_a)$  en los diodos de unión P-N. Para obtener estas curvas aplicamos una tensión externa,  $V_a$ , y dejamos pasar el tiempo suficiente para que se alcance el estado estacionario. Entonces medimos la corriente que atraviesa el dispositivo. Obtenemos así un punto de la característica  $I=f(V_a)$ . Modificando el valor de  $V_a$  y repitiendo el proceso obtendríamos todos los puntos de la curva característica. Este es un procedimiento experimental. Sin embargo, la solución de la ecuación de continuidad nos permite deducir esta curva a partir de la estructura interna y del modelo usado en la *figura 2.3*. Al admitir que la tensión aplicada en los terminales externos,  $V_a$ , aparece en los bordes de la zona de transición se obtiene una expresión exponencial de la forma

$$I = I_s \left( e^{q \cdot V_a / KT} - 1 \right) \quad [2.9]$$

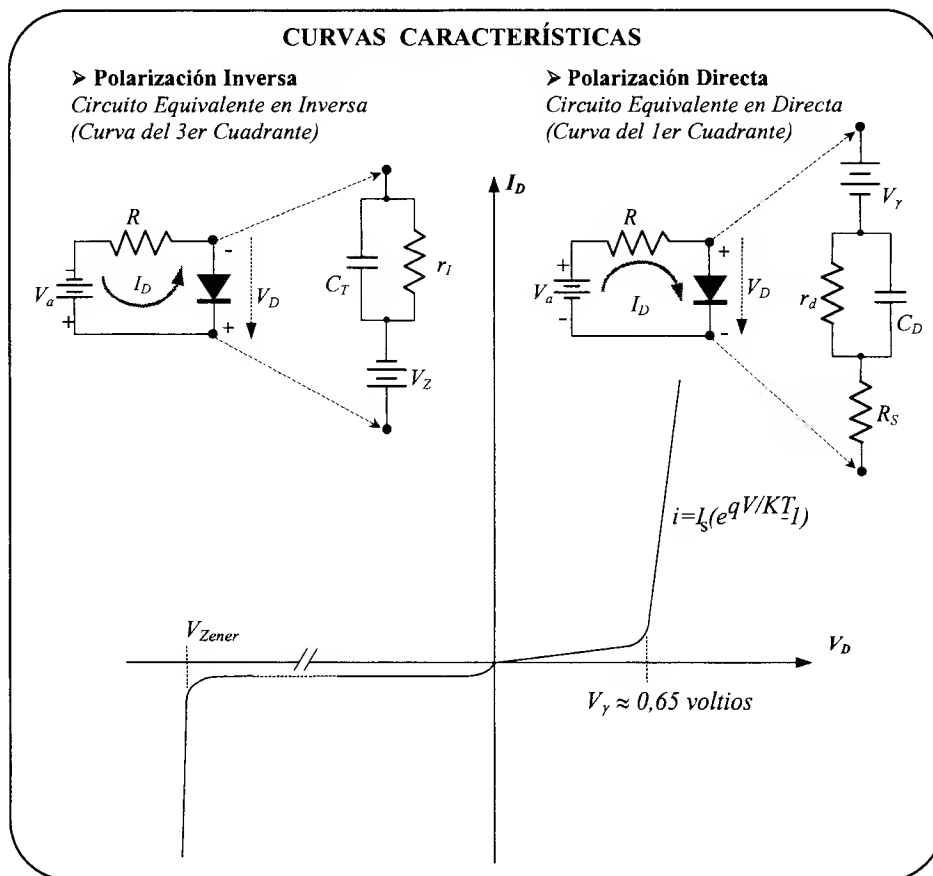
siendo  $K$  la constante de Boltzman,  $T$  la temperatura e  $I_s$  la corriente inversa función de los parámetros eléctricos de la estructura. Para valores altos de  $V_a$  que polarizan al diodo en sentido directo domina el término exponencial

$$I = I_s \cdot e^{q \cdot V_a / KT} \quad [2.10]$$

En caso de polarización inversa el término exponencial desaparece y queda

$$I = - I_s \quad [2.11]$$

Este es el límite del modelo de Shockley, pero la existencia de procesos no considerados al plantear este modelo y en particular los fenómenos de multiplicación por avalancha y efecto túnel, hacen que la característica real en polarización inversa difiera de la ideal en el sentido de que al llegar a un cierto valor de tensión inversa,  $V_a = V_{zener}$ , el dispositivo entra en conducción brusca sólo limitada por las resistencias externas (*figura 2.4*). Sea cual fuere el conjunto de procesos que ocurren en la estructura interna y cuya manifestación estática son las *curvas características*, lo que nos interesa a partir de ahora es el comportamiento del dispositivo como elemento de circuito.



**Figura 2.4.** Característica externa aproximada de un diodo, incluyendo la tensión de despegue,  $V_Y$ , y la zona exponencial en polarización directa (primer cuadrante) y la zona de corte y la ruptura zener en la región de polarización inversa. En la parte superior izquierda se muestra al diodo polarizado en inversa (correspondiente a la curva del tercer cuadrante) y su circuito equivalente incluyendo la tensión zener,  $V_Z$ , la capacidad de transición,  $C_T$ , y la resistencia inversa de alto valor,  $r_i$ . En la parte de la derecha de la figura se muestra el diodo en polarización directa y su circuito equivalente, incluyendo la capacidad de difusión,  $C_D$ , la pila que simula la tensión de despegue,  $V_Y$ , y la resistencia de fugas,  $R_S$  y la resistencia del diodo en directa,  $r_d$ , de bajo valor.

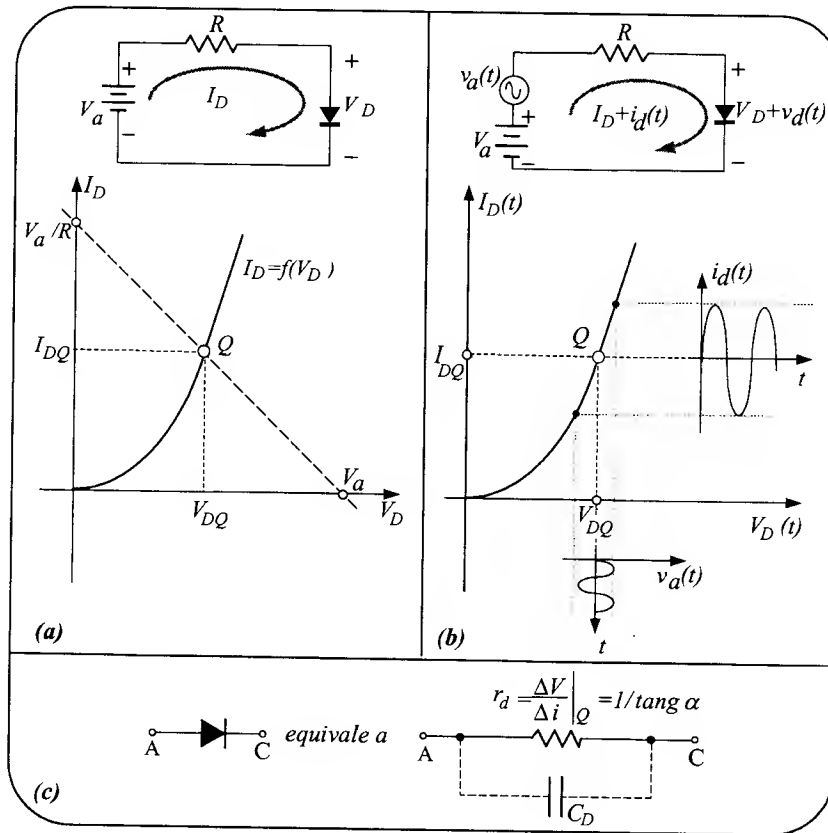
Es decir, le vamos a aplicar un conjunto de valores de tensión entre sus extremos y queremos saber cómo responde modelando su comportamiento en términos de resistencias, condensadores y generadores de tensión.

Cuando usamos un diodo como elemento de circuito empezamos polarizando la unión en

sentido directo o inverso mediante la aplicación de un potencial externo constante,  $V_a$ , y una resistencia  $R$  para limitar la corriente, de acuerdo con el esquema de la *figura 2.5.a*. Obtenemos así un punto de funcionamiento estático  $Q(V_{DQ}, I_{DQ})$ , resultado de la solución gráfica del sistema de ecuaciones:

$$V_a = V_D + R \cdot I_D \quad \text{Recta de Carga (impuesta por el circuito externo)} \quad [2.12]$$

$$I_D = f(V_D) \quad \text{Curva Característica (propia del diodo)} \quad [2.13]$$



**Figura 2.5.** Polarización y comportamiento para pequeñas señales. (a) Punto de trabajo (punto  $Q$ ) resultado de la intersección de la recta de carga,  $V_a = V_D + R \cdot I_D$ , y la curva característica  $I_D = f(V_D)$ . (b) Comportamiento dinámico para pequeñas señales. El punto de trabajo realiza pequeñas oscilaciones en torno al punto de funcionamiento estático ( $Q$ ) y el diodo se comporta como una pequeña resistencia,  $r_d$ . (c) Esta resistencia dinámica es la inversa de la tangente a la curva característica.

Es decir, el punto de funcionamiento estático es el resultado de la intersección de la *recta de carga* con la *curva característica*. La recta de carga la forzamos nosotros desde el exterior mediante la pila  $V_a$  y la resistencia de polarización,  $R$ . En cambio, la curva característica es consecuencia de la estructura interna de la unión P-N. El único punto de trabajo posible es aquel que cumple ambas condiciones. Ese es el punto  $Q$ .

Vamos a estudiar el comportamiento dinámico correspondiente a situaciones en las que la tensión de polarización es función del tiempo. Nos vamos a limitar a pequeñas señales donde  $v(t) = V_a + v_a(t)$  siendo  $v_a(t)$  pequeña. Este caso corresponde al comportamiento lineal de la unión, tal como se ilustra en la *figura 2.5.b*. El diodo se comporta como una pequeña resistencia lineal,  $r_d$ , dando lugar al primer circuito equivalente para pequeñas señales.

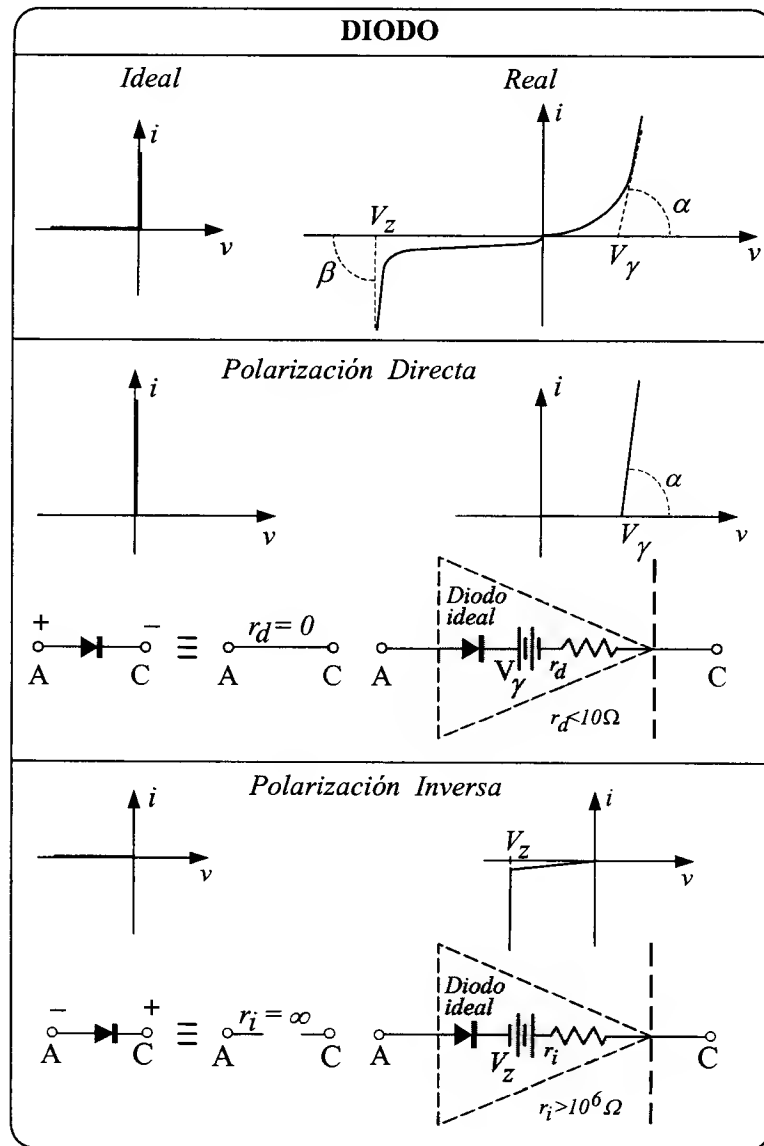
Este modelo hay que completarlo añadiéndole la capacidad  $C_D$  que aparece al resolver la ecuación de difusión dependiente del tiempo y que no vamos a considerar aquí. De la misma manera habría que añadir las resistencias de fugas que tampoco vamos a considerar.

La *figura 2.6* resume la caracterización del diodo como elemento de circuito completando el modelo para pequeñas señales con el modelo para continua (condiciones estáticas) que consta simplemente de un diodo ideal en serie con una pila y con una resistencia dinámica pequeña para el caso de polarización directa y muy grande para el caso de polarización inversa tal como comentamos anteriormente. En continua o para muy bajas frecuencias no aparecen efectos dinámicos y por eso hemos podido eliminar todas las capacidades obteniéndose un circuito equivalente que es el resultado de aproximar por tramos la característica tensión-corriente.

Tal como se ve en la figura el diodo real se separa del diodo ideal en que:

- a) No empieza a conducir en  $V = 0$  sino en  $V = V_\gamma$  (tensión de despegue).
- b) La característica en directa se puede aproximar por una recta, pero la pendiente no es infinita:  $\tan \alpha = 1/r_d$ . Es decir, el diodo cuando conduce, no es un cortocircuito sino que presenta una pequeña resistencia.
- c) El diodo ideal en inversa era un circuito abierto, sin embargo el diodo real en inversa presenta una conductividad muy baja, o sea una resistencia muy alta, pero finita. Además, si seguimos aumentando la polarización inversa, al llegar a un cierto valor,  $V_{zener}$ , la unión P-N entra en zona de ruptura por multiplicación por avalancha o por efecto zener-túnel. Cuando esta ruptura

no es deseada hay que evitarla pero en ciertas ocasiones se busca a propósito para conseguir una función terminal distinta, usada en referencias de tensión y en circuitos reguladores o bien para "sujetar" la tensión de un circuito digital a un cierto valor constante.



**Figura 2.6.** Resumen de los distintos modelos usados para aproximar el comportamiento del diodo, tanto en polarización directa como inversa.

## 2.4. DIODOS EN CONMUTACIÓN.

Toda la electrónica digital está basada en el tratamiento de señales que sólo pueden tener dos valores (por ejemplo cero ó cinco voltios) asociados a los valores lógicos  $V(0) = 0$  voltios y  $V(1) = +5$  voltios. En este caso el diodo opera entre *corte* (polarización en inversa con  $V_a = 0$  voltios) y *activa* (polarización directa con  $V_a = +5$  voltios) y nos interesa caracterizar estos estados y sus transiciones. Es decir, es importante conocer la respuesta de la unión ante saltos en la tensión de polarización porque estos procesos limitan la velocidad de conmutación de un circuito digital. Para comprender los procesos que ocurren en los transitorios de conmutación introducimos el *modelo de control por cargas*.

Volvamos al modelo físico simplificado de la *figura 2.3* y supongamos que la región P está mucho más impurificada que la N de modo que sólo es significativa la componente de corriente asociada a los portadores minoritarios en N, es decir, a  $J_p$  (en N). La ecuación de difusión para huecos en N es:

$$D_p \frac{\partial^2 p_I}{\partial x^2} = \frac{\partial p_I}{\partial t} + \frac{p_I}{\tau_p} \quad [2.14]$$

siendo  $p_I = p_n - p_{n0}$  el exceso de concentración resultado de la inyección desde la otra zona donde son mayoritarios los huecos. La concentración total es  $p_n$  y su valor en situaciones de equilibrio, a temperatura constante y sin inyección de carga,  $p_{n0}$ .

Conocida  $p_I$  de esta ecuación, la densidad de corriente de huecos será:

$$J_p = -q \cdot D_p \frac{\partial p_I}{\partial x} \quad [2.15]$$

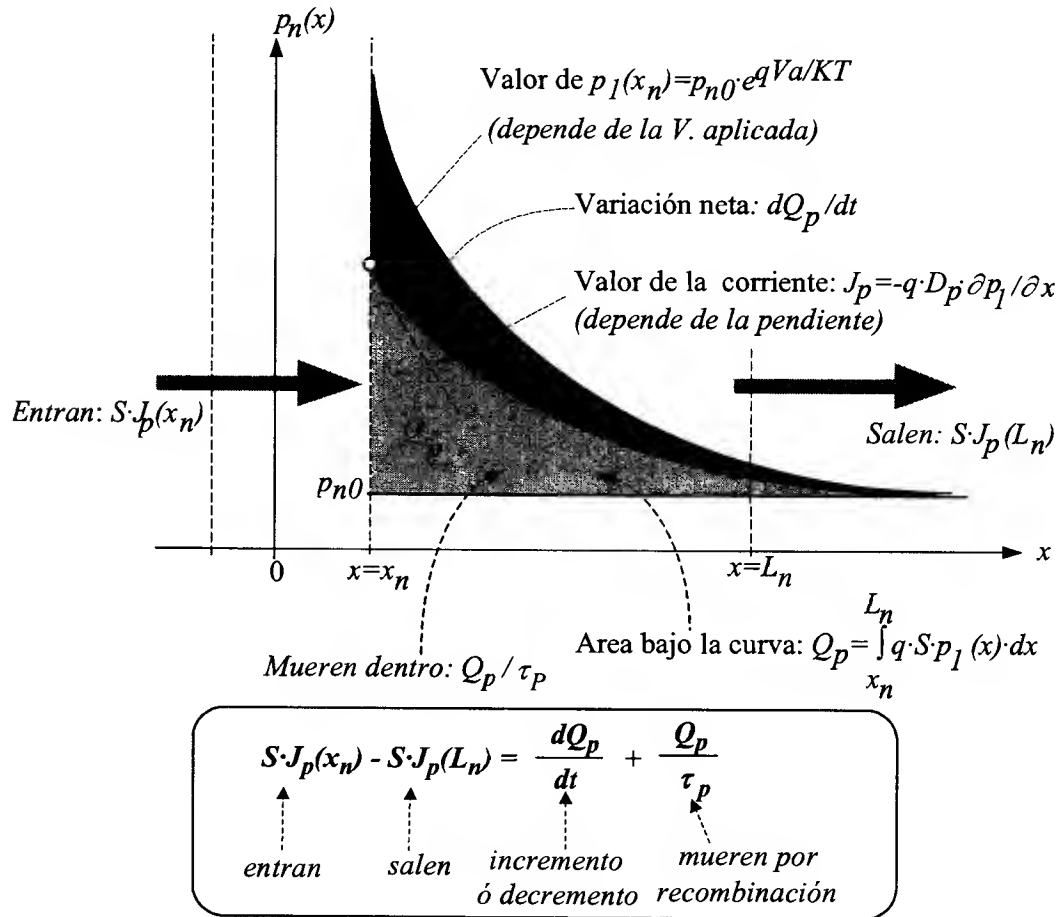
Obsérvese que la tensión aplicada,  $V_a$ , fija el *valor del exceso de concentración de huecos* en la frontera con la zona de transición, suponiendo que es despreciable la caída de tensión en la zona neutra, tal como se ilustra en la *figura 2.7*.

$$p_I(x_n) = p_{n0} \left( e^{q \cdot V_a / KT} - 1 \right) \quad [2.16]$$

En cambio, la corriente que atraviesa el diodo no depende del valor absoluto del exceso de



concentración,  $p_I(x_n)$ , sino de su derivada  $\frac{\partial p_I}{\partial x}$  tal como expresa la ecuación [2.15] y se ilustra en la figura 2.7.



**Figura 2.7.** Ilustración del principio de conservación de la carga en la zona neutra de una unión P-N.

El camino analítico normal ahora consistiría en resolver las ecuaciones [2.15] y [2.16] con las condiciones de contorno adecuadas. Sin embargo, para estudiar el comportamiento de los diodos de unión P-N en electrónica digital no necesitamos conocer la solución exacta y nos basta con poder evaluar los momentos en los que se produce la conmutación y los retardos correspondientes desde que dimos la orden de conmutar cambiando el valor y el signo de la tensión aplicada en sus extremos.

Estos retardos se deben esencialmente a la necesidad de extraer la carga almacenada en la unión. En polarización directa esta carga se almacena esencialmente como exceso de portadores en las zonas neutras. En polarización inversa, la carga se almacena esencialmente en las zonas de transición. Por eso es posible simplificar el análisis viendo sólo lo que pasa con la carga.

Recordemos que estamos usando un diodo con impurificación muy asimétrica y por consiguiente sólo necesitamos estudiar la variación de la carga en la zona neutra de la región N, desde la frontera con la zona de transición,  $x_n$ , hasta la longitud de difusión,  $L_n$ , donde prácticamente ya no hay exceso de concentración sobre el valor de equilibrio ( $p_n(L_n) \approx p_{n0}$  y  $p_I(L_n) \approx 0$ ).

Para obtener la expresión de la variación temporal de la carga de huecos en exceso,  $Q_p$ , almacenada en la zona neutra integramos el exceso,  $p_I(x)$ , entre  $x_n$  y  $L_n$ , es decir calculamos el área bajo la curva de exceso de concentración, (figura 2.7), tras multiplicar por el área transversal del diodo,  $S$ , y por la carga elemental,  $q$ .

$$Q_p = \int_{x_n}^{L_n} q \cdot S \cdot p_I(x) \cdot dx \quad [2.17]$$

Esta carga varía cumpliendo la ecuación de continuidad [2.14] que es la expresión del *principio de conservación de la carga* que vamos a introducir de forma cualitativa. La diferencia entre la carga que entra a la zona neutra por  $x_n$  en la unidad de tiempo,  $S \cdot J_p(x_n)$ , menos la que sale por  $L_n$  en el mismo tiempo,  $S \cdot J_p(L_n)$ , es igual a la que se pierde por recombinación,  $\frac{Q_p}{\tau_p}$ , más  $\frac{dQ_p}{dt}$ , que representa el incremento o disminución en la carga almacenada.

$$S \cdot J_p(x_n) - S \cdot J_p(L_n) = \frac{dQ_p}{dt} + \frac{Q_p}{\tau_p} \quad [2.18]$$

La diferencia, por unidad de tiempo, entre la corriente de huecos que entran en la región neutra de la zona N y los que salen por el contacto metálico es igual a la velocidad con que cambia la carga almacenada más la velocidad con que desaparecen por recombinación.

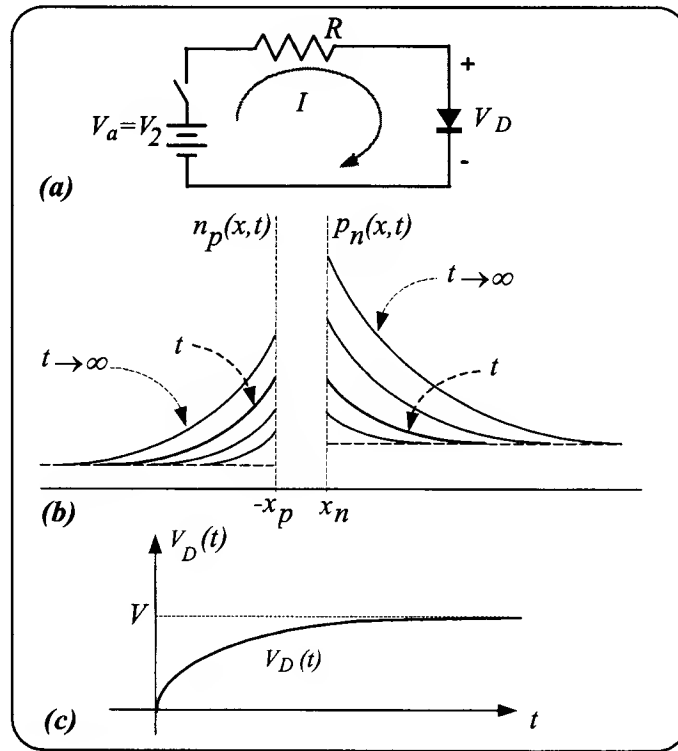
Desgraciadamente, de esta relación tan sencilla entre la carga y la corriente no podemos obtener otra equivalente entre la tensión y la corriente porque el punto de trabajo,  $Q$ , no depende sólo de la tensión, salvo en condiciones estáticas. Estudiaremos pues la relación entre la tensión aplicada y la

corriente para algunos casos asociados al proceso de conmutación entre dos estados estables.

### 2.4.1. Transitorio de Cierre

Supongamos un diodo en serie con una resistencia al que aplicamos de pronto una tensión externa suficiente para polarizarlo en sentido directo y veamos cómo se modifican las concentraciones en el intervalo de tiempo previo al alcance de una situación estacionaria en la que la tensión en los extremos del diodo alcanza un valor constante. La *figura 2.8* ilustra el proceso. Al aplicar una tensión de polarización en directa,  $V_a = V_2$ , suficientemente mayor que la tensión de despegue,  $V_\gamma$ , la caída en la resistencia de polarización marca la corriente,  $I$ , que atraviesa el diodo.

$$I = \frac{V_2 - V_D}{R} \approx \frac{V_2}{R} \quad [2.19]$$



**Figura 2.8.** Transitorio de cierre. (a) Circuito. (b) Crecimiento del exceso de portadores en zona neutra al pasar el tiempo. (c) Transitorio y estacionario de  $V_D(t)$ .

Esto produce un cambio brusco en el valor de la pendiente de la concentración de portadores en la frontera de la zona de transición que pasa de cero a :

$$-\left. \frac{\partial pI(x)}{\partial x} \right|_{x=x_n} = \frac{V_2/R}{qSD_p} = \frac{V_2}{qSRD_p} \quad [2.20]$$

Obsérvese que este cambio en el valor de la pendiente es muy rápido ya que basta con inyectar unos pocos portadores. La nueva corriente va aumentando la carga almacenada en la zona neutra, subiendo la curva del exceso de concentración tal como se muestra en la *figura 2.8.b*, hasta que se alcanza el estado estacionario. Entonces,

$$I = I_S \left( e^{q \cdot V_D / K \cdot T} - 1 \right)$$

y por consiguiente: 
$$V_D = \frac{KT}{q} \ln \left( \frac{I}{I_S} + 1 \right) \quad [2.21]$$

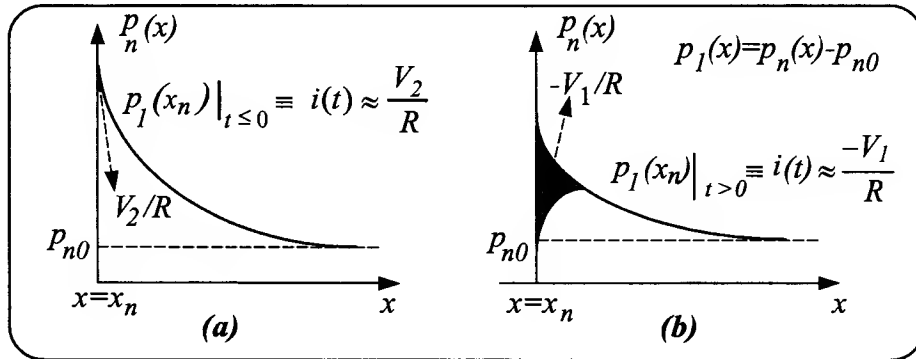
## 2.4.2. Transitorio de Corte

Partimos ahora de la situación anterior, en la que el diodo está polarizado en sentido directo mediante la tensión  $V_a=V_2$ , de la que caía  $V_D$  en la unión en estado estacionario. Para realizar una conmutación, cambiamos el valor de la tensión aplicada en magnitud y signo. Este cambio en la tensión aplicada,  $V_a=V_1$ , terminará polarizando a la unión en sentido inverso en condiciones estacionarias, cuando pase algún tiempo. Ahora bien, lo que nos interesa ahora no es el estado estacionario sino conocer el comportamiento transitorio. Es decir, ver cómo se alcanza el estado estacionario y cuánto tiempo consume el proceso de conmutación, porque de este parámetro dependerá la frecuencia máxima a la cual pueden usarse los circuitos digitales construidos a partir de estas uniones P-N. La pregunta que nos hacemos es *¿cómo se alcanza el nuevo estado estacionario de corte?*.

Desde luego, lo primero que cambia es la magnitud y el signo de la corriente pues para ello basta con *cambiar la pendiente de la concentración* y esto sólo exige mover unos pocos portadores, tal como se ilustra en la *figura 2.9*.

Al invertir la corriente cambia la pendiente pero el diodo sigue en polarización directa. La corriente inversa se gasta en compensar el exceso de carga y la tensión no cambia de signo hasta que

no pasamos por  $p_I(x_n) = 0$ . La disminución de  $Q_p$ , proporcional a la integral entre  $x_n$  y  $L_n$  del exceso,  $p_I(x)$ , se debe a la extracción y a la recombinación (término  $Q_p / \tau_p$ ).



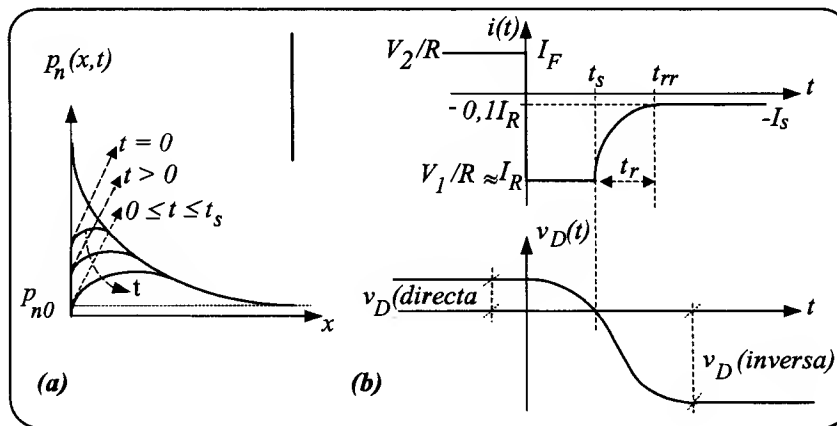
**Figura 2.9.** Cambio de pendiente en el exceso de portadores en la frontera de la zona de transición como consecuencia del paso de polarización directa a inversa. **(a)** Situación estacionaria en polarización directa. **(b)** Situación transitoria tras el cambio de la tensión externa.

Aunque se haya invertido la corriente (pendiente), el diodo sigue en polarización directa hasta que  $p_n(x_n)$  no pasa por debajo del valor de equilibrio,  $p_{n0}$ . Tras la inversión, los huecos almacenados son insuficientes para mantener la pendiente de  $p(x_n)$  (corriente), por lo que no se puede mantener el ritmo de extracción  $I_I = V_I / R$  pudiendo entonces ocurrir dos cosas:

- Si estamos alimentando por un generador de corriente aumentará la tensión inversa hasta introducir a la unión en una zona de ruptura.
- Si  $V_I$  es superior a la tensión de ruptura y el generador no es ideal, la corriente inversa cae hasta el valor de saturación,  $-I_S$ , tal como se muestra en el esquema de la figura 2.10.

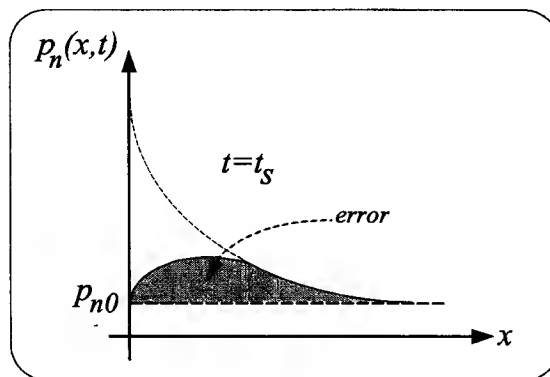
El tiempo que tarda en invertirse la tensión, después de haberse invertido la corriente se llama *tiempo de retardo por almacenamiento*,  $t_s$ . Lógicamente cuanto mayor sea la corriente inversa,  $-I_I$ , y menor el tiempo de vida media de los portadores, menor será también este tiempo de retardo,  $t_s$ . Su valor se puede estimar usando el modelo de control por carga y suponiendo diodos largos en los que la zona N es mucho mayor que la longitud de difusión para huecos, por lo que podemos despreciar el valor de la corriente de difusión en las proximidades del contacto metálico, de [2.18] obtenemos,

$$J_p(L_n) \approx 0 \Rightarrow S \cdot J(x_n) = \frac{dQ_p}{dt} + \frac{Q_p}{\tau_p} \quad [2.22]$$



**Figura 2.10.** Transitorio de corte. (a) Forma del exceso de concentración,  $p_n(x)$ . (b) Corrientes y tensiones terminales en las que se representa de forma expandida el transitorio para mostrar el significado físico del proceso.

Si observamos la *figura 2.11* vemos que despreciando el área matizada podemos calcular  $t_s$  a partir del instante  $t$  en el que la carga pasa por cero,  $Q_p(t) = 0$ .



**Figura 2.11.** Detalle del error producido al calcular  $t_s$ .

Por definición,  $t_s$ , es el tiempo necesario para eliminar toda la carga almacenada haciendo igual a cero el valor de  $Q_p$  en  $t = t_s$ .

Antes de conmutar, en  $t < 0$ , como estamos en situaciones estacionarias,  $\frac{dQ_p}{dt} = 0$  y, por

consiguiente, la ecuación [2.22] nos da:  $I_2 = \frac{Q_p(0^-)}{\tau_p}$

Después, tan pronto como se realiza la conmutación, en  $t \geq 0$ , aplicando de nuevo la ecuación [2.22], tenemos

$$-I_1 = \frac{dQ_p}{dt} + \frac{Q_p}{\tau_p} \Rightarrow Q_p(t) = -I_1 \cdot \tau_p + K \cdot e^{-\frac{t}{\tau_p}} \quad [2.23]$$

$$\text{para } t=0, \quad Q_p = I_2 \cdot \tau_p = -I_1 \cdot \tau_p + K \Rightarrow K = \tau_p(I_1 + I_2) \quad [2.24]$$

luego la evolución temporal de la carga seguirá una expresión exponencial del tipo,

$$Q_p(t) = -I_1 \cdot \tau_p + \tau_p \cdot (I_1 + I_2) \cdot e^{-\frac{t}{\tau_p}} \quad [2.25]$$

Recordando finalmente que  $t_s$  se definía como el tiempo en el cual  $Q_p(t_s) = 0$ , hacemos  $Q_p(t_s) = 0$  en la expresión anterior y obtenemos:

$$t_s = \tau_p \cdot \ln \left( 1 + \frac{I_1}{I_2} \right) \quad [2.26]$$

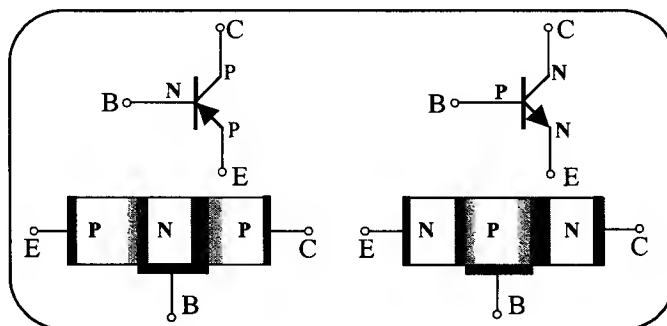
## 2.5. TRANSISTORES BIPOLARES

Un transistor bipolar es un dispositivo de tres terminales, llamados *base*, *emisor* y *colector* constituido por dos uniones P-N con una zona (base) común. Hay, por consiguiente, dos tipos de transistores bipolares PNP y NPN que se representan por los símbolos de la *figura 2.12*.

El nombre de *transistor* procede de la compresión de las palabras inglesas *Transfer Resistor* que describen su principio de funcionamiento consistente en conseguir transferir un mismo nivel de corriente desde una impedancia baja (unión base-emisor) a una alta (unión base-colector), con la consiguiente ganancia en tensión y potencia.

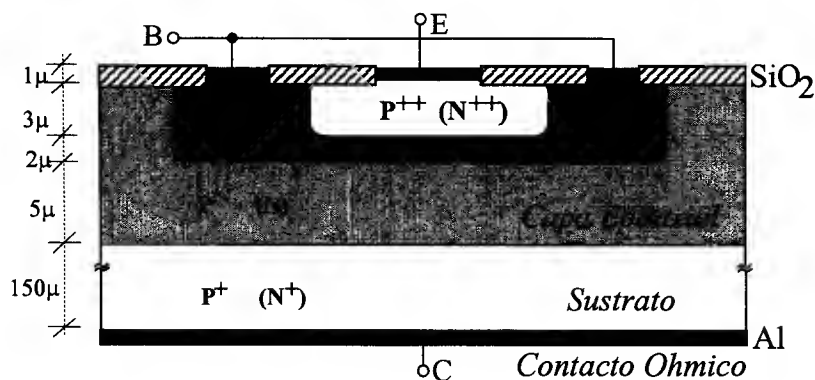
Se le llama *bipolar* porque la conducción se realiza en las dos bandas, es decir, haciendo intervenir a los electrones y a los huecos tal como ocurría en los diodos de unión y en contraposición a lo que ocurre con los transistores de efecto campo que veremos más adelante en los que la

conducción se debe sólo a los portadores mayoritarios. Estos dispositivos que conducen usando un único tipo de portador se llaman *unipolares*.



**Figura 2.12.** Símbolo y modelo simplificado de la estructura interna en transistores bipolares PNP y NPN.

De entre las distintas estructuras físicas posibles, la tecnología actual es del tipo *planar-epitaxial* con un campo gradual en la base que ayuda, por deriva, a los mecanismos de conducción por difusión. La *figura 2.13* muestra la forma de esta estructura. Para el caso NPN, por ejemplo, se parte de una sustrato  $N^+$  y sobre él se hace crecer una capa epitaxial tipo N. Después se oxida de nuevo la superficie del Silicio y se abren ventanas para proceder a la difusión de nuevas impurezas del tipo de las del sustrato y formar así una segunda unión. Finalmente, se abren ventanas para metalizar, estableciendo así contactos óhmicos sobre las dos difusiones que van a dar lugar a los terminales externos de base y emisor, respectivamente.

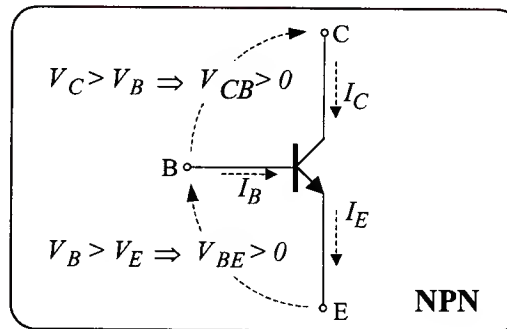


**Figura 2.13.** Estructura física de un transistor bipolar en tecnología planar.



Obsérvese que el terminal de colector está unido al sustrato. La *región de colector* está así constituida por una *capa epitaxial* y un *sustrato*, ambos de impurezas del mismo tipo, pero este con una concentración mucho más elevada que aquella.

Dado que un transistor consta de dos uniones, existen cuatro modos de funcionamiento posibles según que cada una de estas dos uniones se polaricen en sentido directo o inverso. La tabla de la *figura 2.14* resume estos modos y el signo de las tensiones para un transistor NPN. Para un transistor PNP estos signos deben invertirse.



Unión Emisor-Base	Unión Base-Colector	Modo de funcionamiento
<b>Polarización Directa</b>	<b>Polarización Directa</b>	<b>Saturación</b>
Polarización Directa	Polarización Inversa	Activo Normal ó directo
Polarización Inversa	Polarización Directa	Activo Inverso
<b>Polarización Inversa</b>	<b>Polarización Inversa</b>	<b>Corte</b>

**Figura 2.14.** Modos de funcionamiento.

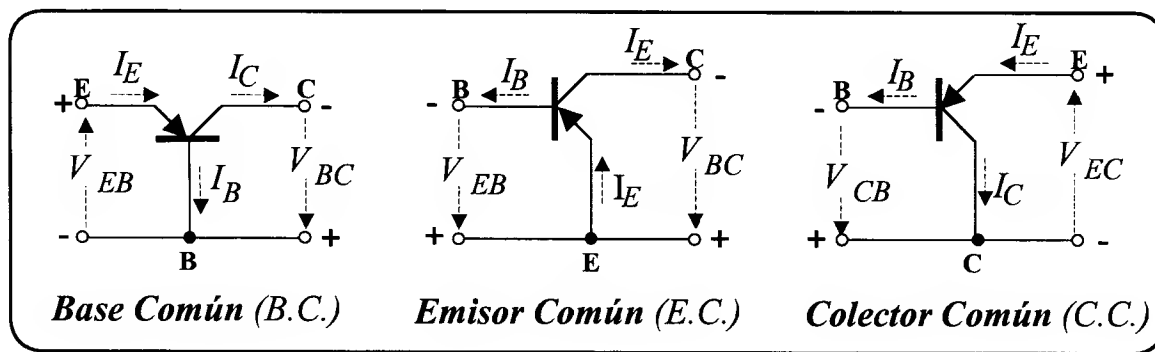
Los modos de saturación y corte son los propios de la Electrónica Digital, en general, donde el transistor actúa como un conmutador (analógico) que pasa de resistencia muy baja (cerrado, saturación) a resistencia muy alta (abierto, corte) de acuerdo con el valor de la tensión en la base. Aquí aparece el soporte electrónico de la acción "inversor". La salida (colector) de un transistor se conecta como entrada a la base de otro transistor, de forma que, cuando el primero está en corte, el segundo está en saturación y viceversa. Si el corte se asocia al estado lógico  $V(“1”)$  y la saturación al estado lógico  $V(“0”)$ , aparece una estructura que soporta la función lógica de complementación.

El modo de funcionamiento lineal y propio de la Electrónica Analógica es *el modo activo normal* o directo en el que la unión base-emisor está polarizada en sentido directo y la unión base-colector en sentido inverso. En estas condiciones el transistor es un *amplificador*. Las variaciones en la tensión base-emisor dan lugar a variaciones en la corriente de emisor ( $I_E$ ), pero sobre una impedancia mucho mayor. Este es el principio de funcionamiento del transistor bipolar: dos uniones P-N con un *área común tan estrecha* que los portadores que inyecta una de las uniones son recogidos por la otra prácticamente sin tener tiempo de recombinarse.

### 2.5.1. Diagrama de Corrientes: Estudio Cualitativo del Transistor Bipolar.

Supongamos ahora que tenemos un transistor PNP polarizado externamente con dos pilas que hacen que la unión de emisor está polarizada en sentido directo y la unión de colector en sentido inverso. Tenemos así una configuración en *modo activo normal*.

Como el transistor es un dispositivo de tres terminales, al ser estudiado como elemento de circuito, como un cuadripolo, uno de los terminales debe ser común a la entrada y a la salida dando así lugar a tres configuraciones básicas: *base común*, *emisor común* y *colector común* (figura 2.15).



**Figura 2.15.** Configuraciones básicas de un transistor bipolar: (B.C.: Base Común, E.C.: Emisor Común y C.C.: Colector Común).

Haremos el estudio del diagrama de corrientes y la caracterización estática sobre la configuración en *base común*.

El tratamiento general y analítico del efecto transistor queda fuera del alcance de este tema que

sólo pretende introducir los conocimientos sobre dispositivos electrónicos imprescindibles para estudiar las familias lógicas. Por eso nos limitaremos a estudiar los *efectos de primer orden* usando un *modelo unidimensional* en los siguientes pasos:

- a) Análisis cualitativo (diagrama de corrientes).
- b) Solución de la ecuación de difusión para portadores minoritarios en la zona neutra de la región de base.
- c) Circuitos equivalentes.

Con estos conocimientos mínimos ya podremos pasar al estudio del transistor en *corte* y *saturación* y los *transitorios* asociados al cambio de estado, que son los procesos más usuales en *electrónica digital*.

La *figura 2.16* muestra el esquema de un transistor PNP polarizado en modo activo normal, el perfil de impurezas suponiendo uniones abruptas, el diagrama de bandas de energía y la forma aproximada de los excesos de concentración de minoritarios, consecuencia de la yuxtaposición de las concentraciones correspondientes a las dos uniones.

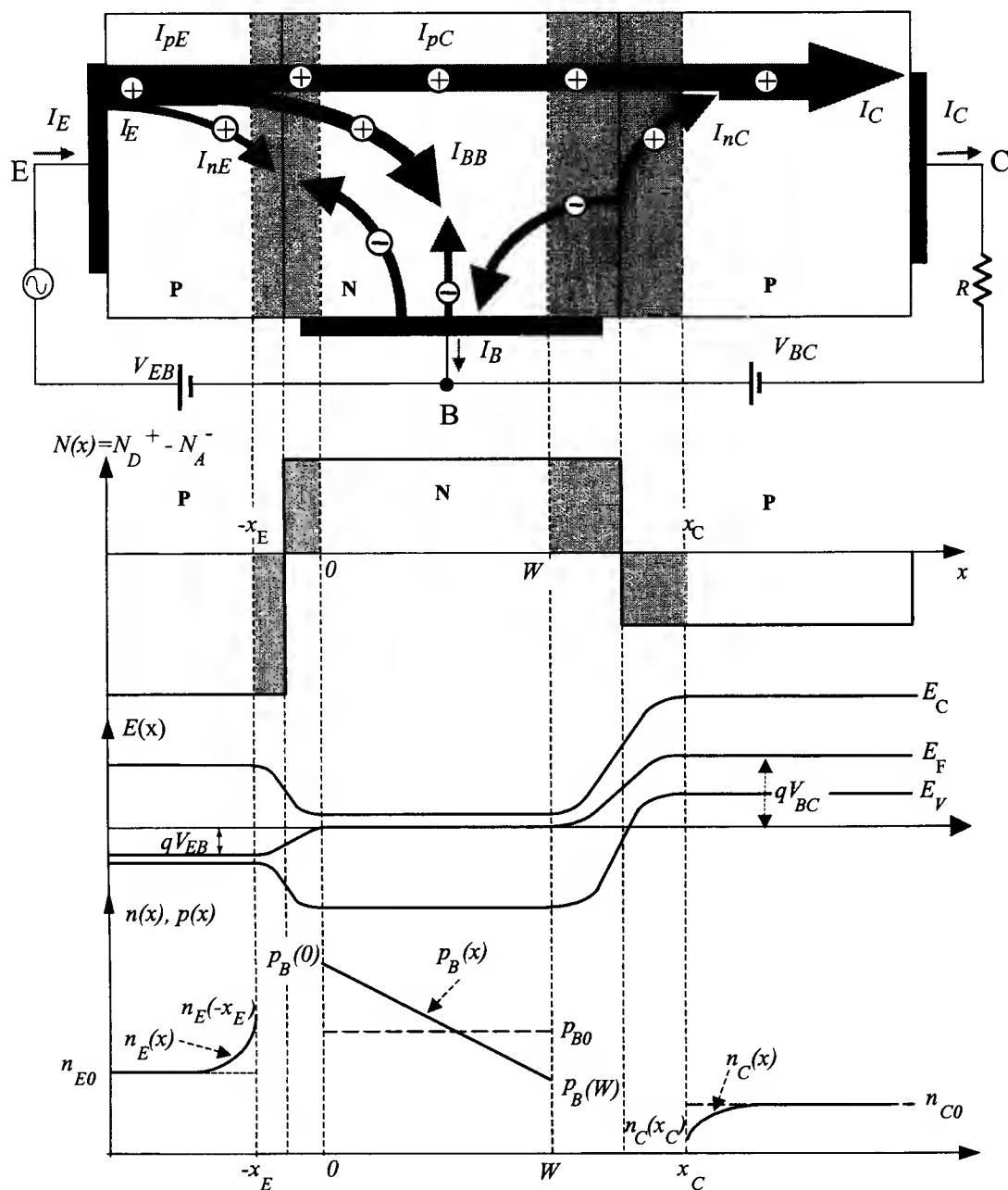
Obsérvese que en cada unión existe una zona de transición, más ancha en la unión de colector que en la de emisor debido a que aquella está polarizada en sentido inverso y que el colector está mucho menos impurificado que el emisor. La anchura efectiva de la base,  $W_B$ , es la longitud comprendida entre los bordes de ambas regiones de transición por el lado de la base.

Empecemos describiendo el *diagrama de corrientes*. La unión de emisor está polarizada en sentido directo por lo que inyecta un gran número de huecos en la base,  $I_{pE}$ . La otra componente de la corriente de emisor,  $I_{nE}$ , representa los huecos que se consumen en la frontera de la zona de transición para recombinarse con los electrones que inyecta la base en el emisor. Así pues:

$$I_E = I_{nE} + I_{pE} \quad [2.27]$$

Sigamos ahora el camino de los portadores que se inyectan en la base, gran parte de estos la atraviesan sin tener tiempo de recombinarse, dando lugar a la componente de la corriente de huecos en el colector,  $I_{pC}$ . Sin embargo, algunos de estos portadores se recombinan durante el tránsito de la zona neutra de base dando lugar a una pequeña componente de la corriente de base,  $I_{BB}$ .

$$I_{pC} = I_{pE} - I_{BB} \quad [2.28]$$



**Figura 2.16.** Esquema del transistor bipolar con el diagrama de corrientes, la distribución de carga y el efecto de la polarización ( $V_{EB}$  y  $V_{BC}$ ) sobre las barreras de potencial y la distribución del exceso de concentración en las proximidades de las uniones.

La otra componente de la corriente de colector es la propia de una unión P-N polarizada en sentido inverso,  $I_{nC}$  (extracción de portadores desde una región en la que son minoritarios). Es decir:

$$I_C = I_{pC} + I_{nC} \quad [2.29]$$

Finalmente, la corriente en el terminal de base posee las tres componentes que hemos descrito anteriormente: inyección en el emisor,  $I_{BE} = I_{nE}$ , extracción del colector,  $I_{BC} = -I_{nC}$  y aporte de electrones para recombinarse con los huecos que atraviesan la zona neutra de base,  $I_{BB}$ . Así pues:

$$I_B = I_{BE} + I_{BB} + I_{BC} = I_{nE} + I_{BB} - I_{nC} \quad [2.30]$$

Globalmente las tres corrientes deben cumplir el principio de conservación de la carga, de forma que:

$$I_E = I_B + I_C \quad [2.31]$$

Obsérvese de nuevo que la acción transistor se debe a la estrechez de la región de base que hace que por la unión de colector, polarizada en sentido inverso, pase una corriente alta, prácticamente igual a la inyectada por la unión de emisor que está muy cerca. Si alejáramos ambas uniones, aumentando  $W_B$ , se perdería el efecto transistor y tendríamos sólo dos diodos en oposición (*figura 2.17.a*). El efecto de la proximidad de las uniones lo representamos en términos de elementos de circuito introduciendo un generador de corriente dependiente de  $I_E$  y prácticamente igual a esta ( $\alpha I_E$ , siendo  $\alpha \approx 1$ ) en paralelo con la unión de colector (*figura 2.17.b*).

Hay tres parámetros que caracterizan la calidad de un transistor y que en un caso ideal deberían valer la unidad para la configuración en base común:

a) *Eficiencia de emisor:*  $\gamma = \frac{I_{pE}}{I_E} = \frac{I_{pE}}{I_{pE} + I_{nE}} \leq 1$ . Mide la participación de la inyección de

emisor en la corriente total. En un caso ideal, se aproxima a la unidad, debido a que, por construcción, la región de emisor está mucho más impurificada que la base.

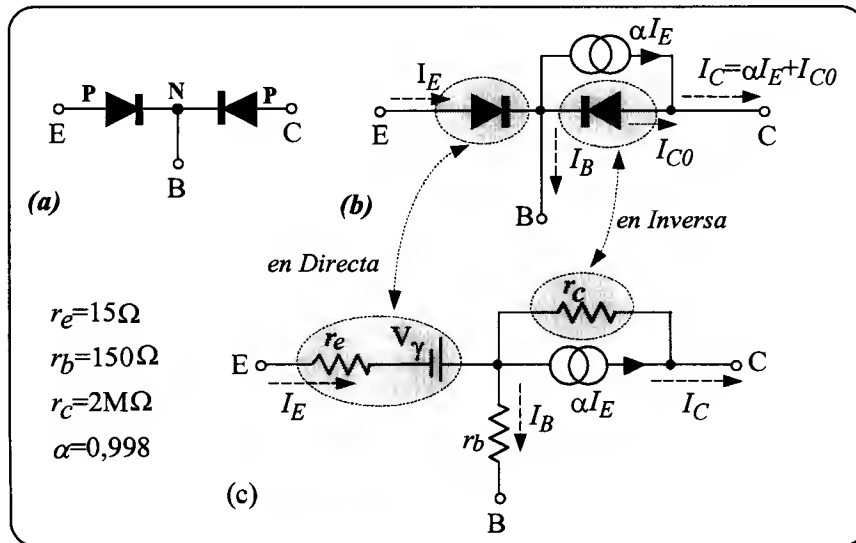
b) *Factor de transporte:*  $\beta^* = \frac{I_{pC}}{I_{pE}} \leq 1$ . Mide los minoritarios que atraviesan la zona neutra de

base sin recombinarse. En un caso ideal de base muy estrecha y con la conducción ayudada por un campo eléctrico (base gradual, transistor de deriva)  $\beta^*$  sería igual a 1.

- c) *Factor de multiplicación de colector:*  $\alpha^* = \frac{I_C}{I_{pC}}$ . Da una medida de la contribución de la corriente de huecos procedentes del emisor a la corriente total del colector.

Estos tres parámetros se combinan para dar lugar a la ganancia en corriente en base común,  $\alpha$ , que será:

$$\alpha = h_{FB} = \frac{I_C}{I_E} = \frac{I_C}{I_{pC}} \cdot \frac{I_{pC}}{I_{pE}} \cdot \frac{I_{pE}}{I_E} = \alpha^* \cdot \beta^* \cdot \gamma \approx 1 \quad [2.32]$$



**Figura 2.17.** Nacimiento progresivo del modelo del transistor bipolar considerado como elemento de circuito. En (a) se ilustra la pérdida del efecto transistor al alejar las dos uniones. Sólo nos quedan dos diodos en oposición. En (b) aparece el efecto transistor por la estrechez de la región de base. Ahora, la corriente del emisor alcanza prácticamente toda ( $\alpha \approx 1$ ) al colector y esto se representa a nivel de elementos de circuito poniendo un generador dependiente de corriente en paralelo con el "diodo" de colector. En (c) aparece un circuito equivalente resultado de completar el modelo anterior fenomenológicamente. Es decir, añadiendo aquellos elementos que sabemos que están, como  $r_e$ ,  $r_b$  y  $V_\gamma$ , aunque no hayan aparecido en la parcela de teoría usada para obtener la etapa anterior del modelo. Como la unión de emisor está polarizada en sentido directo, nos vamos a la parte correspondiente del modelo de la unión de la figura 1.4 y sustituimos el diodo por  $r_e$  en serie con  $V_\gamma$ . Análogamente, como la unión de colector está en inversa, sustituimos el diodo correspondiente por una resistencia de alto valor,  $r_c$ .

### 2.5.2. Circuitos Equivalentes del Transistor Bipolar

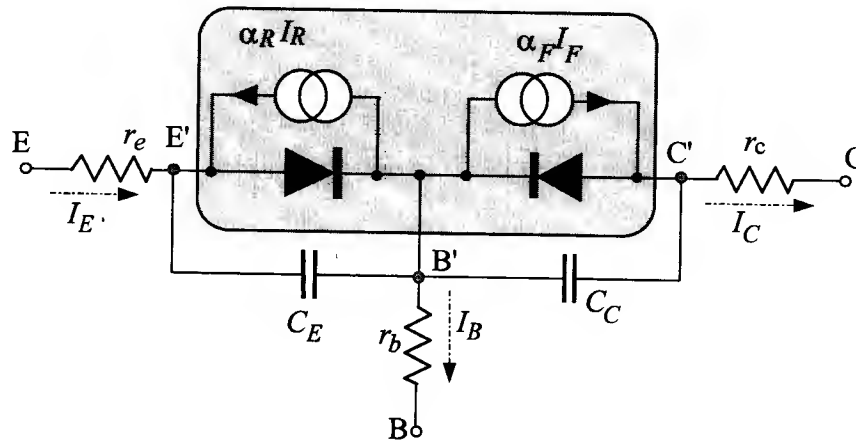
A partir de aquí podemos ya obtener un primer circuito equivalente del transistor construido fenomenológicamente, (figura 2.17.c). El emisor lo sustituimos por el modelo de una unión P-N polarizada en sentido directo, es decir:  $r_e$  y  $V_\gamma$ . En el colector incluimos la alta resistencia de una unión polarizada en inversa ( $r_c$ ) y un generador de corriente dependiente de la de emisor,  $\alpha I_E$ .

La siguiente etapa que se ilustra en la figura 2.18 es el *Modelo de Ebers-Moll* con el que se completa la caracterización del transistor bipolar en *condiciones estáticas*. Es decir, cuando se aplican las tensiones de polarización ( $V_{EB}$  y  $V_{CB}$ ) se espera el tiempo necesario para que se alcancen las situaciones estacionarias y entonces se mide el valor que han alcanzado las corrientes,  $I_C$ ,  $I_E$  é  $I_B$ , obteniendo:

$I_E$  = función de ( $V_{BE}$ ,  $V_{CB}$ ): *Característica de entrada*.

$I_C$  = función de ( $V_{BE}$ ,  $V_{CB}$ ): *Característica de salida*.

Siendo:  $I_B = I_E - I_C$



**Figura 2.18.** Modelo de Ebers-Moll resultado de considerar la simetría de la situación. Ahora, partiendo de la figura 2.17.c, se ha añadido en el emisor un nuevo generador dependiente de la corriente del colector y se ha modificado levemente la nomenclatura introduciendo los subíndices R (reverse) y F (forward).

Obsérvese que en un caso general los valores del potencial en las uniones,  $V_{BE}$  y  $V_{BC}$ , podrán

ser positivos o negativos, dando lugar a polarizaciones *directas* o *inversas* y consecuentemente a los cuatro modos de funcionamiento que hemos comentado anteriormente (activo normal, activo inverso, saturación ó corte).

El modelo de *Ebers-Moll* aparece analíticamente al resolver la ecuación de difusión, pero aquí lo vamos a introducir fenomenológicamente.

Como las tensiones  $V_{EB}$  y  $V_{CB}$  pueden tomar cualquier valor, la corriente en el terminal de colector siempre se podrá escribir como la correspondiente a la unión de colector  $\left[ I_{CS} \left( e^{qV_{CB}/KT} - 1 \right) \right]$  más la inyectada desde el emisor a través de la base  $\left[ \alpha_F \cdot I_F = \alpha_F I_{ES} \left( e^{qV_{EB}/KT} - 1 \right) \right]$ . Análogamente, la corriente de emisor siempre quedará descrita por una expresión que incluya la correspondiente a la unión de emisor  $\left[ I_{ES} \left( e^{qV_{EB}/KT} - 1 \right) \right]$  más la que pueda proceder de la inyección desde el colector  $\left[ \alpha_R \cdot I_R = \alpha_R \cdot I_{CS} \left( e^{qV_{CB}/KT} - 1 \right) \right]$ . Por consiguiente, en general:

$$I_E = I_F - \alpha_R \cdot I_R = I_{ES} \left( e^{qV_{EB}/KT} - 1 \right) - \alpha_R \cdot I_{CS} \left( e^{qV_{CB}/KT} - 1 \right) \quad [2.33]$$

$$I_C = \alpha_F \cdot I_F - I_R = \alpha_F \cdot I_{ES} \left( e^{qV_{EB}/KT} - 1 \right) - I_{CS} \left( e^{qV_{CB}/KT} - 1 \right) \quad [2.34]$$

Cumplíendose para la corriente de base,

$$I_B = I_E - I_C = I_F (1 - \alpha_F) + I_R (1 - \alpha_R) \quad [2.35]$$

Estas ecuaciones se corresponden con las ecuaciones de los *nudos* E', C' y B' de la *figura 2.18*. Lo único que hemos hecho ha sido sumar algebraicamente y de acuerdo con el criterio de signos de cuadripolos, las corrientes que entran en E' (expresión [2.33]), en C' (expresión [2.34] y en B' (expresión (2.35)). De nuevo, por razonamientos fenomenológicos, hemos introducido las resistencias  $r_e$ ,  $r_c$  y  $r_b$  correspondientes a la resistencia óhmica de las zonas neutras de material semiconductor existentes desde las regiones de transición hasta los terminales externos. Por los mismos razonamientos se han introducido dos capacidades,  $C_E$  y  $C_C$ , que si bien no actúan a baja frecuencia, serán dominantes cuando el transistor se integre en circuitos dinámicos que operen a altas frecuencias.

Recordemos finalmente que este modelo de *Ebers-Moll* es general y vale para cualquier



combinación de valores y signos en las tensiones de polarización de las uniones ( $V_{EB}$  y  $V_{CB}$ ). Ahora bien, en cada *situación real* el modelo se *simplificará*. Por ejemplo, en el modo activo normal ( $V_{EB} > 0$  y  $V_{CB} < 0$  y grande en valor absoluto) queda:

$$I_E = I_{ES} e^{\frac{qV_{EB}}{KT}} + \alpha_R I_{CS} \approx I_F$$

$$I_C = \alpha_F I_{ES} e^{\frac{qV_{EB}}{KT}} + I_{CS} \approx \alpha_F I_F$$

con lo que obtenemos de nuevo el modelo simplificado de la *figura 2.17.b*. En emisor hay una unión directa dominada por el término exponencial y en colector sólo existe la parte de esa corriente ( $\alpha_F I_F$ ,  $\alpha_F < 1$ ) recogida por la unión de colector que está en inversa, ya que su propia corriente inversa,  $I_{CS}$ , es despreciable en términos relativos.

### 2.5.3. Curvas Características y Regiones de Funcionamiento de los Transistores Bipolares

Las curvas características representan la relación entre los valores de tensiones y corrientes en los extremos de un transistor. Así, si el transistor está en configuración de emisor común, sus características de entrada,  $I_B = f(V_{BE})$ , muestran la relación entre la corriente de base,  $I_B$ , y la tensión base-emisor,  $V_{BE}$ , cuando la tensión colector-emisor,  $V_{CE}$ , se mantiene como parámetro constante, tal como se muestra en la *figura 2.19.b*. Análogamente, las características de salida,  $I_C = f(V_{CE})$ , muestran la relación entre la corriente de colector,  $I_C$ , y la tensión entre colector y emisor,  $V_{CE}$ , para  $I_B$  constante, (*figura 2.19.c*).

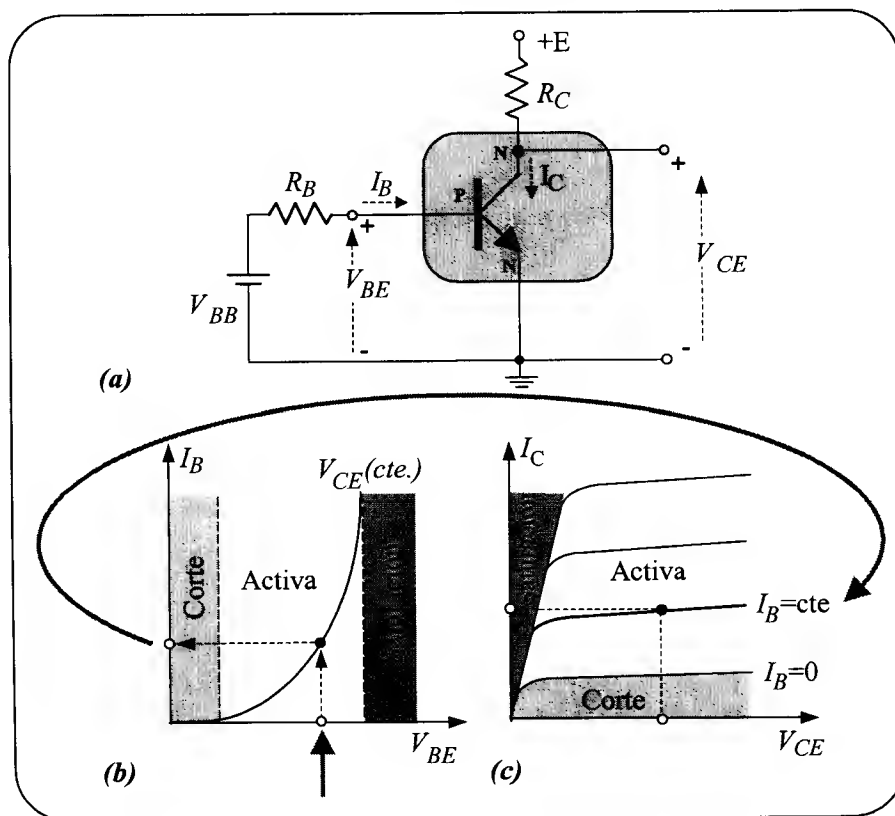
Sobre ambas características se distinguen las tres zonas de funcionamiento: *corte*, *activa* y *saturación*. En *corte*, el transistor no conduce, porque la unión de emisor no alcanza la polarización directa,  $V_{BE} = 0,6$  voltios. Por consiguiente  $I_B = 0$  y en la característica de salida la región de corte corresponde a la zona que está por debajo de la curva correspondiente a  $I_B = 0$ . Si aumentamos la tensión base-emisor, el transistor permanece en zona activa, la unión base-emisor conduce como un diodo polarizado en directa y el colector recoge esta corriente. La característica de entrada se parece a una exponencial y las características de salida son rectas casi paralelas al eje de abscisas. Es decir, para  $V_{BE} > V_{BE(on)}$ ,  $I_B$  aumenta con  $V_{BE}$ . Para valores positivos y constantes de  $I_B$ ,  $I_C$  aumenta con  $V_{CE}$  y

mantiene una relación sencilla con  $I_B$

$$I_C = \beta I_B + I_{CO}(\beta + 1) \quad [2.36]$$

siendo

$$\beta = \frac{\alpha}{1 - \alpha} \quad [2.37]$$



**Figura 2.19.** Curvas características y regiones de funcionamiento en transistores bipolares en la configuración de emisor-común. (a) Circuito de medida. (b) Característica de entrada con sus tres zonas. (c) Características de salida, mostrando también las tres zonas. La línea de puntos muestra la relación causal entre los valores de ambas características. Un valor de  $V_{BE}$  produce uno de  $I_B$ . Este fija una característica ( $I_B = \text{cte.}$ ) en la salida, a donde hay que ir para ver la relación en  $I_C$  y  $V_{CE}$ .

Las características de entrada se parecen a las del diodo estudiadas en el apartado anterior ya que en definitiva la unión base-emisor de un transistor es una unión P-N que en el modo activo,

cuando el transistor actúa como amplificador, está polarizada en sentido directo.

En efecto, recordemos la expresión de la corriente de colector en función de la de emisor,  $I_E$  y la corriente inversa de la unión de colector ( $I_{C0}$ )

$$I_C = \alpha \cdot I_E + I_{C0} \Rightarrow I_E = \frac{I_C - I_{C0}}{\alpha} \quad [2.38]$$

La corriente de base será:

$$I_B = I_E - I_C = \frac{I_C - I_{C0}}{\alpha} - I_C = I_C \cdot \frac{1 - \alpha}{\alpha} - \frac{I_{C0}}{\alpha} \quad [2.39]$$

$$I_C = \frac{\alpha}{1 - \alpha} I_B + \frac{1}{1 - \alpha} I_{C0} = \beta \cdot I_B + (\beta + 1) I_{C0} \quad [2.40]$$

Este parámetro ( $\beta$ ) es representativo de la ganancia en corriente del transistor y tiene un valor entre 150 y 500. Cuando se refiere sólo al funcionamiento en continua se le suele llamar  $h_{FE}$ .

Llega un momento al final de la zona activa en la que si seguimos aumentando la tensión en la unión base-emisor, el transistor entra en zona de *saturación*. En las características de entrada se alcanza para valores de  $V_{BE}$  del orden de 0,8 ó 0,9 *voltios*, de modo que el intervalo de zona activa es pequeño (0,2 ó 0,3 *voltios*), a partir de 0,6 (tensión de despegue). En las características de salida representa la región de la izquierda en la que  $V_{CE} = V_{CE} (sat) \approx 0,2$  *voltios*, de forma que ambas uniones están polarizadas en sentido directo. Ahora el transistor ofrece una resistencia muy pequeña.

## 2.6. EL TRANSISTOR BIPOLAR EN CORTE Y SATURACIÓN.

Hasta ahora hemos visto el principio de funcionamiento del transistor bipolar y su comportamiento en condiciones estáticas a través de un conjunto de modelos sencillos y de sus curvas características. El siguiente paso es el estudio del comportamiento dinámico. Para ello se siguen dos pasos:

- a) *Polarizar al transistor (elección del punto Q).*
- b) *Superponer al punto Q una señal externa variable con el tiempo,  $v_a(t)$ , con dos situaciones de interés:*
  - b.1) La señal externa,  $v_a(t)$ , es pequeña de forma que el transistor no sale nunca de zona activa.

Esta es la forma de trabajo propia de la electrónica analógica y de la *lógica no saturada* (familia ECL) en electrónica digital.

- b.2)** La señal externa,  $v_a(t)$ , toma uno de dos valores posibles y el salto es grande, de forma que puede llevar al transistor de corte a saturación y viceversa. El transistor actúa como un conmutador inversor dando lugar a toda la *lógica saturada* que domina la electrónica digital.

Vamos a ver primero cómo se polariza un transistor y después estudiaremos su comportamiento dinámico, haciendo énfasis en la caracterización de los estados de corte y saturación y en el cálculo de los tiempos de conmutación.

Polarizar un transistor bipolar es elegir un punto de funcionamiento estático (punto  $Q$ ), con dos valores para  $V_{CEQ}$  é  $I_{CQ}$  y hacer que el transistor se sitúe en ese punto. Para ello se usan dos fuentes externas y dos resistencias,  $R_C$  y  $R_B$  y resolvemos gráficamente el sistema de ecuaciones de la malla de salida. La *figura 2.20* muestra el esquema de *polarización*. Supongamos que en la entrada  $V_{BB}$  y  $R_B$  permanecen constantes polarizando la unión de emisor en zona activa con un cierto valor de  $I_B$ . Este valor de  $I_B = cte$  fija una curva característica de salida concreta,  $I_C = f(V_{CE})_{I_B = cte}$ . En estas condiciones vamos a analizar la malla de salida. Mirando hacia el transistor, vemos una tensión,  $V_{CE}$ , y una corriente,  $I_C$ , que están relacionadas por la curva característica que hemos seleccionado al fijar  $I_B$  en la base.

$$I_C = f(V_{CE}), \text{ para } I_B = cte \quad [2.41]$$

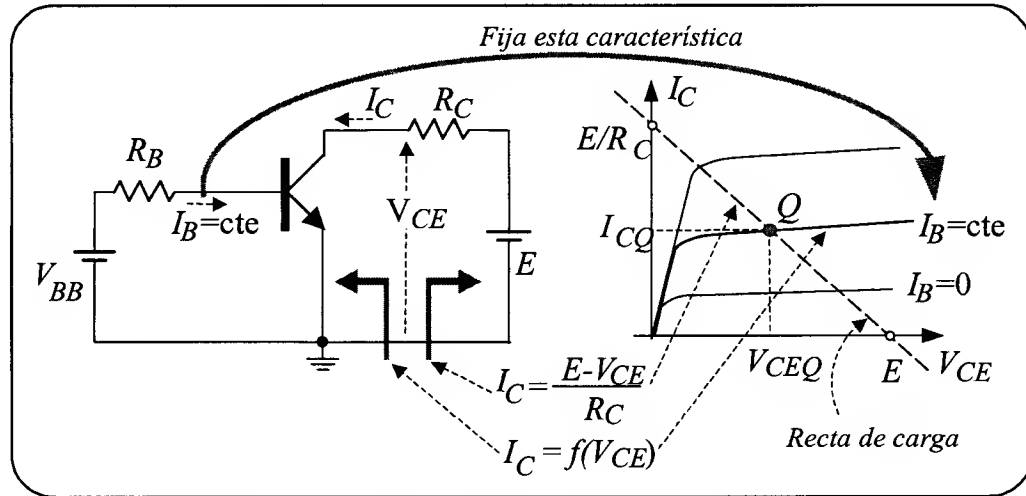
Por otro lado, mirando hacia la derecha vemos  $R_C$  y la pila  $E$ , de forma que debe cumplirse

$$V_{CE} = E - I_C R_C \text{ (Recta de Carga)} \quad [2.42]$$

Así, el punto de funcionamiento estático ( $Q$ ) es aquel que cumple ambas condiciones a la vez ([41], [42]). La solución de este sistema de dos ecuaciones con dos incógnitas se obtiene de forma gráfica, encontrando la intersección de la recta de carga,  $V_{CE} = E - I_C R_C$ , con la curva característica,  $I_C = f(V_{CE})$ . Obsérvese que el punto  $Q$  se puede controlar variando  $I_B$  (a través de  $R_B$  y  $V_{BB}$  en la entrada) o variando la recta de carga (a través de  $E$  y  $R_C$  en la salida).

Si ya tenemos polarizado el transistor, veamos ahora su comportamiento dinámico (punto b.1 y b.2). En el comportamiento para pequeñas señales (b.1), al valor de polarización se superpone una

señal externa  $v_a(t)$  tal que  $|v_a(t)| \ll |E|$ , como se ilustra en la *figura 2.21.a*. En este caso, la unión de emisor nunca sale de zona activa y el transistor actúa como un amplificador lineal. Su estudio es propio de la electrónica analógica y no vamos a desarrollarlo aquí.

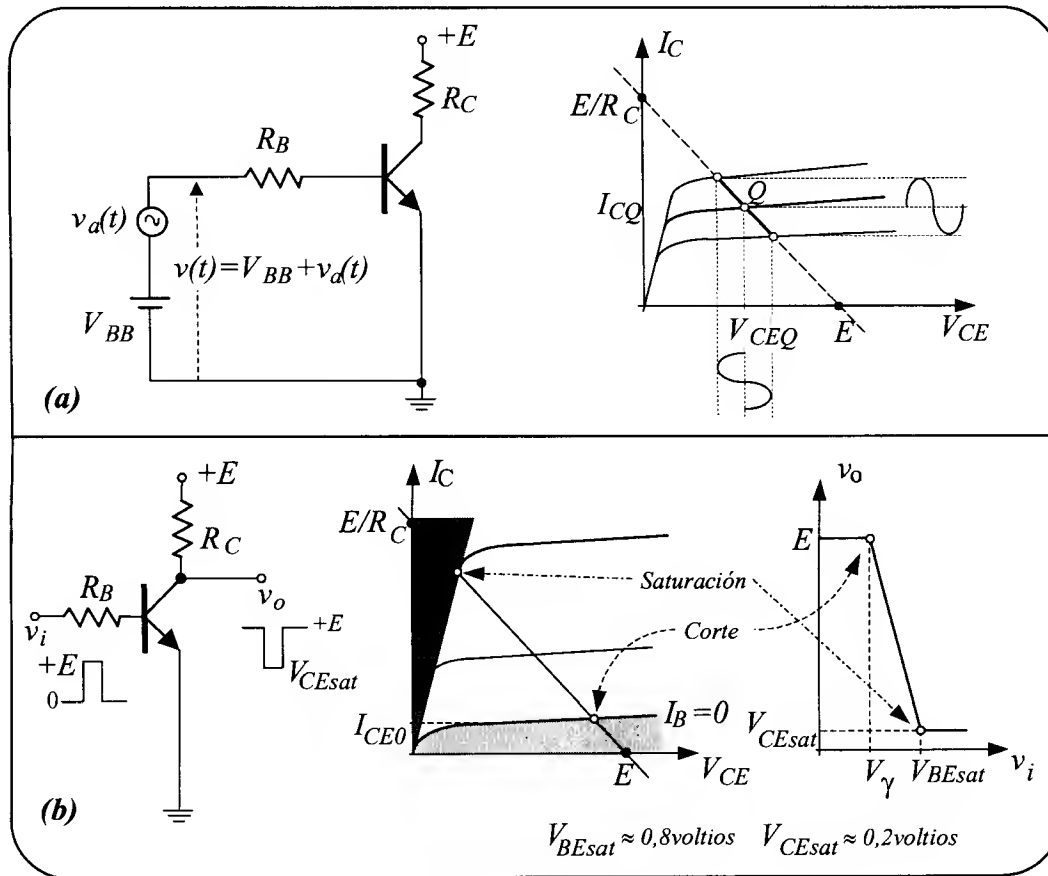


**Figura 2.20.** Polarización del transistor bipolar

Si ya tenemos polarizado el transistor, veamos ahora su comportamiento dinámico. En el comportamiento para pequeñas señales, al valor de polarización se superpone una señal externa  $v_a(t)$  tal que  $|v_a(t)| \ll |E|$ , como se ilustra en la *figura 2.21.a*. En este caso, la unión de emisor nunca sale de zona activa y el transistor actúa como un amplificador lineal. Su estudio es propio de la electrónica analógica y no vamos a desarrollarlo aquí.

Consideremos ahora la situación de la *figura 2.21.b*. En este caso el transistor actúa como conmutador porque la tensión externa  $v_a(t)$  salta entre dos valores extremos (0 y +E, por ejemplo) que hacen que el punto Q recorra toda la zona activa pasando de *corte* a *saturación* y viceversa. Decimos entonces que el transistor actúa como un *inversor* que sintetiza la característica de transferencia de la *figura 2.1*, tal como propusimos al comienzo del tema.

El problema en este caso es caracterizar los estados estacionarios por una parte y estudiar después los transitorios en la transición entre estos estados. La *figura 2.21.b* ilustra el circuito mínimo de un inversor que pasa de corte a saturación dependiendo del estado de la señal en base.



**Figura 2.21.** Comportamiento dinámico. (a) Pequeñas señales. (b) Corte y saturación: circuito inversor.

Recordemos que la relación entre las corrientes (ecuaciones [2.38] y [2.40]) era:

$$I_C = \alpha \cdot I_E + I_{C0} \quad \text{y} \quad I_C = \beta \cdot I_B + (\beta + 1)I_{C0}$$

Así, hasta que la tensión en la base no supere un valor umbral,  $V_\gamma \approx 0.5\text{voltios}$ , la unión de emisor no conduce y el transistor está al corte. En estas condiciones  $I_B = 0$  y la única corriente que queda en el colector es:

$$I_C = (\beta + 1)I_{C0} = I_{CE0} \quad [2.43]$$

Si la tensión en base sigue aumentando, el transistor atraviesa la zona activa y entra en saturación para  $V_{BE} \geq 0.8V$ . A partir de este momento la tensión de colector queda fijada en un valor

pequeño y constante  $V_{CEsat}=0.2\text{voltage}$ .

El valor de la corriente de saturación depende de la resistencia de carga  $R_C$  y del valor de la pila que hemos usado en polarización,  $E$ . Entonces, los dos estados estables en lógica saturada son:

<b>Corte</b>	{	$\begin{cases} I_C = I_{CE0} \\ V_{CE} \approx E \end{cases}$	{	<b>Saturación</b>	{	$\begin{cases} I_C = I_{Csat} \approx E/R_C \\ V_{CE}=V_{CEsat} \approx 0.2 \text{ V} \end{cases}$
--------------	---	---	---	-------------------	---	--

Para producir estos cambios de estado es necesario que la tensión  $V_{BE}$  pase desde un punto por debajo de  $0,5V$  hasta un punto por encima de  $0,8V$  (figura 2.21.b).

Veamos ahora la caracterización de estos estados desde un punto de vista más físico, aplicando el modelo de Ebers-Moll, que vimos en la figura 2.18.

### 2.6.1. Estado de Corte

En esta situación ambas uniones están polarizadas en sentido inverso y para  $|V_{BE}| \gg \frac{KT}{q}$  y  $|V_{CB}| \gg \frac{KT}{q}$  las exponenciales del modelo de Ebers-Moll (ecuaciones [33] y [34]) desaparecen quedando las expresiones simplificadas

$$I_E = -I_{ES} + \alpha_R \cdot I_{CS} \quad [2.44]$$

$$I_C + \alpha_F \cdot I_{ES} + I_{CS} \quad [2.45]$$

$$\text{Como } \alpha_R \cdot I_{CS} = \alpha_F \cdot I_{ES} \quad [2.46]$$

$$I_E = -I_{ES}(1 - \alpha_F) \quad [2.47]$$

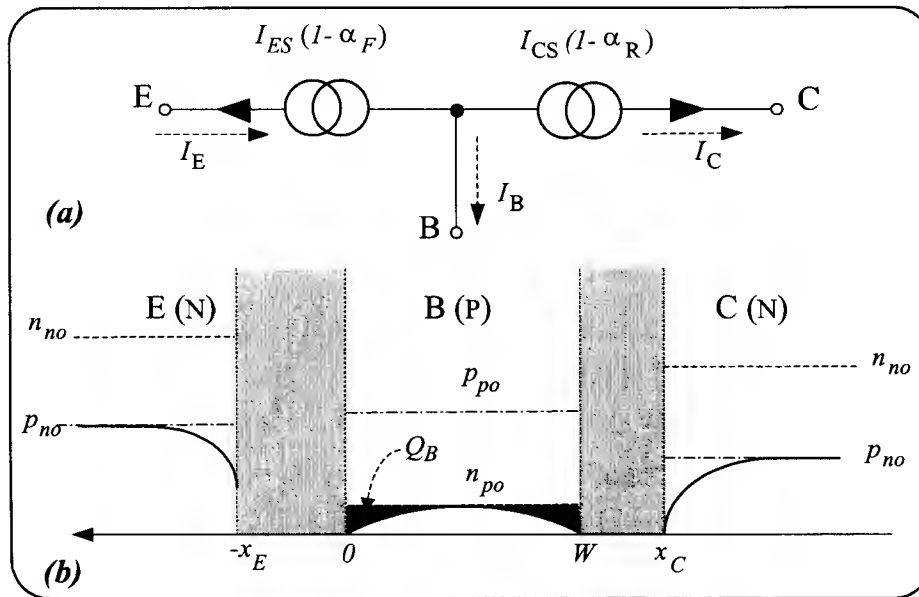
$$I_C = -I_{CS}(1 - \alpha_R) \quad [2.48]$$

$$I_B = I_E - I_C = -I_{ES}(1 - \alpha_F) - I_{CS}(1 - \alpha_R) \quad [2.49]$$

Para transistores de Silicio estas corrientes son prácticamente despreciables, se encuentran en el

rango de los *nanoamperios*, aunque son función de la temperatura pues se deben a la generación térmica de minoritarios. En estado de corte no hay inyección de portadores minoritarios en región de base, sólo la extracción de los generados térmicamente. Las tensiones inversas que se pueden aplicar a ambas uniones están limitadas por los fenómenos de ruptura por avalancha. En cuanto a la carga, la almacenada en la zona neutra de base es despreciable, pero no así la almacenada en las capacidades de transición de las uniones de emisor y colector, que son las responsables del retardo en la conmutación.

La figura 2.22 muestra el modelo de Ebers-Moll simplificado para el estado de corte y la distribución de carga en las proximidades de las zonas de transición. Obsérvese que todas las concentraciones están por debajo de sus valores de equilibrio y que en la zona neutra hay poca carga almacenada ( $Q_B$ ).



**Figura 2.22.** Estado de corte (a) Versión simplificada del modelo de Ebers-Moll. (b) Concentraciones y carga en las proximidades de las zonas de transición.

### 2.6.2. Estado de Saturación

Al polarizar las uniones fuertemente en sentido directo se pone en marcha una serie de procesos transitorios que hacen la transición del estado OFF al ON. El punto  $Q$  atraviesa rápidamente la zona activa y entra en saturación donde se fija en un nuevo estado estacionario. De nuevo los valores de las

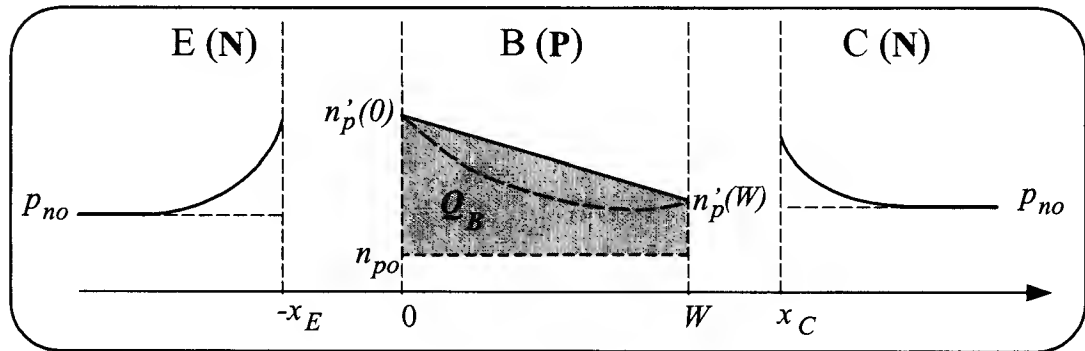


tensiones base-emisor y colector-base son altos  $\left(|V_{EB}| \gg \frac{KT}{q} \text{ y } |V_{CB}| \gg \frac{KT}{q}\right)$ , pero ahora son ambas positivas (en PNP)  $V_E > V_B$  y  $V_C > V_B$  y ambas negativas en NPN.

El transistor ofrece una resistencia muy pequeña (tanto en zona de transición como en regiones neutras). La *figura 2.23* muestra los excesos de concentraciones haciendo énfasis en la zona neutra de base y suponiendo que la polarización directa de la zona de emisor es más fuerte que la de colector. Por eso  $n'_p(0) > n'_p(W)$ . Ahora la carga almacenada,  $Q_B$ , es grande.

$$n'_p(0) = n_{po} \left( e^{qV_{BE}/KT} - 1 \right) \quad [2.50]$$

$$n'_p(W) = n_{po} \left( e^{qV_{CB}/KT} - 1 \right) \quad [2.51]$$



**Figura 2.23.** Estado de saturación. Valor de las concentraciones de carga en exceso en las proximidades de las zonas de transición. Ahora no introducimos la versión correspondiente del modelo de Ebers-Moll porque coincide con la general.

Aplicando de nuevo el modelo de Ebers-Moll obtenemos el valor de las tensiones en saturación en función de las corrientes terminales. Para ello reescribimos las ecuaciones [2.33] y [2.34], quitando el uno en los paréntesis, ya que ahora, al estar ambas uniones polarizadas fuertemente en sentido directo, el término exponencial es dominante y las componentes de corriente inversa ( $-I_{ES}$ ,  $-I_{CS}$ ) son despreciables

$$I_E = I_{ES} \cdot e^{qV_{EB}/KT} - \alpha_R \cdot I_{CS} \cdot e^{qV_{CB}/KT} \quad [2.52]$$

$$I_C = \alpha_F \cdot I_{ES} \cdot e^{qV_{EB}/KT} - I_{CS} e^{qV_{CB}/KT} \quad [2.53]$$

Multiplicando la ecuación [2.53] por  $\alpha_R$  y restándola de [2.52] obtenemos:

$$I_E - \alpha_R \cdot I_C = (1 - \alpha_R \cdot \alpha_F) \cdot I_{ES} \cdot e^{qV_{EB}/KT} \quad [2.54]$$

Despejando  $I_E$  resulta

$$I_E = \alpha_R \cdot I_C + (1 - \alpha_R \cdot \alpha_F) \cdot I_{ES} \cdot e^{qV_{EB}/KT} \quad [2.55]$$

$$\text{luego} \quad I_E = \alpha_R \cdot I_C + I_{E0} \cdot e^{qV_{EB}/KT} \quad [2.56]$$

Análogamente, multiplicando la ecuación [2.52] por  $\alpha_F$ , restándola de la ecuación [2.53] y despejando  $I_C$ , obtenemos:

$$I_C = \alpha_F \cdot I_E - (1 - \alpha_F \cdot \alpha_R) \cdot I_{CS} \cdot e^{qV_{CB}/KT} \quad [2.57]$$

$$I_C = \alpha_F \cdot I_E - I_{C0} \cdot e^{qV_{CB}/KT} \quad [2.58]$$

Con esto ya tenemos los valores de la corriente de emisor,  $I_E$ , y de colector en términos de las tensiones base-emisor y colector-base. Como lo que buscamos es la expresión del valor de la tensión colector-emisor en saturación vamos a despejar  $V_{BE}$  de [2.56] y  $V_{CB}$  de [2.58] obtenemos:

$$V_{BE} = \frac{KT}{q} \ln \frac{I_E - \alpha_R \cdot I_C}{I_{E0}} \quad [2.59]$$

$$V_{CB} = \frac{KT}{q} \ln \frac{\alpha_F \cdot I_E - I_C}{I_{C0}} \quad [2.60]$$

Sustituyendo ahora  $I_E$  por su valor en función de  $I_B$  é  $I_C$  ( $I_E = I_B + I_C$ ) en las ecuaciones [2.59] y [2.60] resulta:

$$V_{BE} = \frac{KT}{q} \ln \frac{I_B + I_C - \alpha_R \cdot I_C}{I_{E0}} = \frac{KT}{q} \ln \frac{I_B + I_C(1 - \alpha_R)}{I_{E0}} \quad [2.61]$$

$$V_{CB} = \frac{KT}{q} \ln \frac{\alpha_F \cdot (I_B + I_C) - I_C}{I_{C0}} = \frac{KT}{q} \ln \frac{\alpha_F I_B + I_C(1 - \alpha_F)}{I_{C0}} \quad [2.62]$$

y, finalmente, restando las expresiones de la tensión base-emisor y de la tensión colector-base, obtenemos la tensión de colector-emisor en saturación

$$\begin{aligned}
 V_{CE}|_{sat} &= V_{CB} - V_{BE} = \frac{KT}{q} \ln \left[ \frac{\alpha_F I_B - I_C (1 - \alpha_F)}{I_{C0}} \cdot \frac{I_{E0}}{I_B + I_C (1 - \alpha_R)} \right] = \\
 &= \frac{KT}{q} \ln \frac{\left( \frac{\alpha_F}{1 - \alpha_F} \cdot \frac{I_B}{I_C} - 1 \right) (1 - \alpha_F) I_{E0}}{I_{C0} \left[ \frac{I_B}{I_C} + (1 - \alpha_R) \right]} \\
 V_{CE}|_{sat} &= \frac{KT}{q} \ln \frac{(1 - \alpha_F) \cdot I_{E0}}{I_{C0}} \cdot \frac{\beta_F \frac{I_B}{I_C} - 1}{\frac{I_B}{I_C} + 1 - \alpha_R}
 \end{aligned}
 \tag{2.63}$$

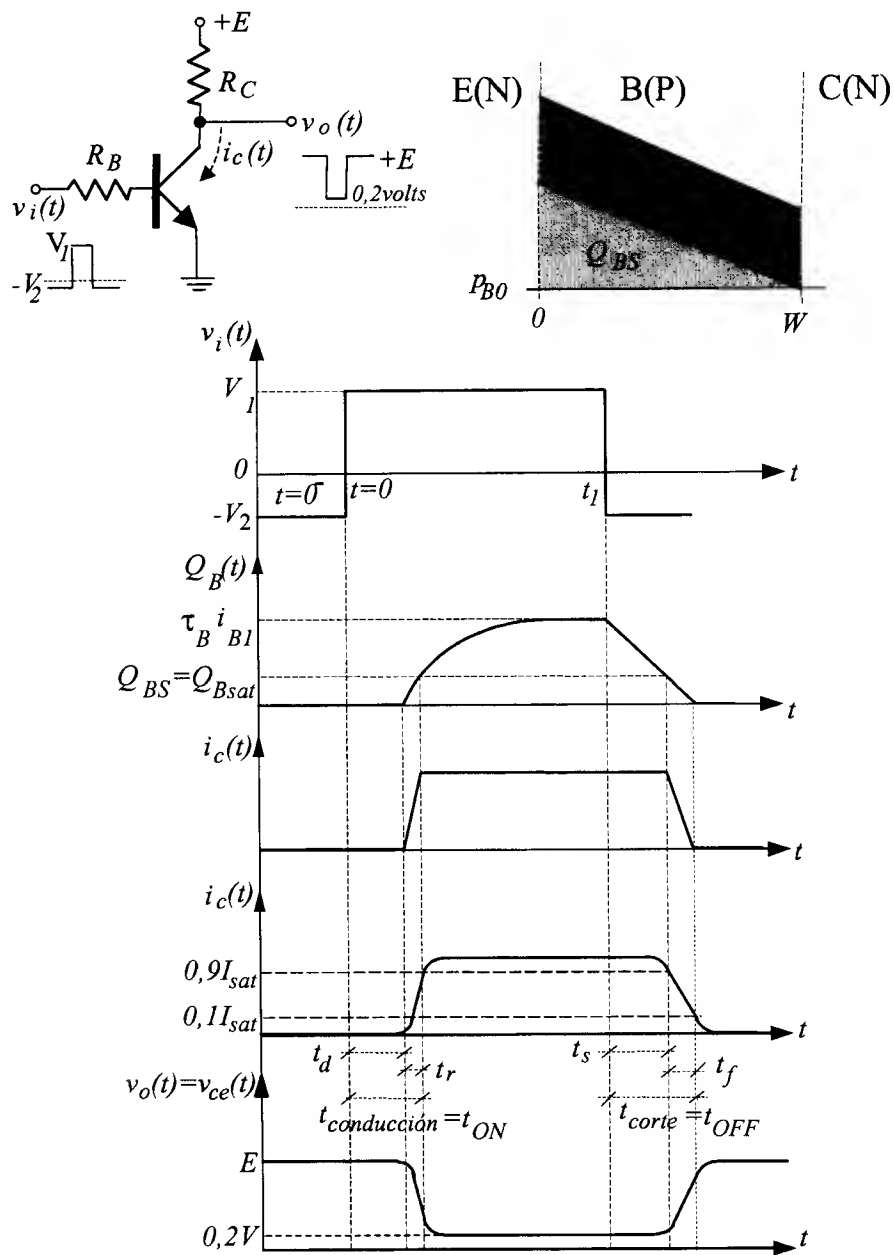
En resumen, en saturación ambas uniones están polarizadas en directa, todas las tensiones entre los terminales son pequeñas y las corrientes están limitadas por las resistencias y las fuentes de alimentación externas. La expresión [2.63] permite el cálculo de  $V_{CE}|_{sat}$  y su dependencia con la temperatura, punto de funcionamiento y parámetros del transistor a través de  $\alpha_R$  y  $\alpha_F$ .

### 2.6.3. Transiciones entre Estados

En condiciones normales de operación los parámetros críticos en conmutación son justo los transitorios del paso de corte a saturación y viceversa y los tiempos de retardo que introducen. Estos retardos se deben, básicamente, a la necesidad de mover las cargas almacenadas en la zona neutra de base y en las zonas de transición.

Consideremos un transistor NPN en configuración de emisor común que inicialmente está al corte y al que le aplicamos en la base un pulso de señal que le obliga a recorrer un ciclo completo: *corte-saturación-corte*. La figura 2.24, muestra primero el circuito, después la distribución de carga en la zona neutra de base, entre las uniones de emisor y colector.

Finalmente, en la parte inferior de la figura se muestra la evolución temporal de los valores de tensión de entrada, carga almacenada, valor aproximado de la corriente de colector, valor más próximo al real en el que se acota una zona de subida lineal y otra de bajada y, finalmente, tensión de salida, que coincide con la tensión colector-emisor y que es una versión invertida y retardada del pulso de entrada.



**Figura 2.24.** Transiciones entre estados. Ciclo **corte-saturación-corte**. (Ver descripción en el texto).

Los distintos tiempos que caracterizan los transitorios son:

$t_d \Rightarrow$  *Tiempo de retardo*: es el tiempo que transcurre desde  $t=0$  (comienzo de la excitación) hasta que se inicia la respuesta.

$t_r \Rightarrow$  *Tiempo de subida*: es el tiempo desde que responde hasta que  $I_C = 0,9I_{Csat}$ .

$t_s \Rightarrow$  *Tiempo de almacenamiento*: desde que comienza la excitación de paso al corte hasta que inicia la disminución de  $I_C$ .

$t_f \Rightarrow$  *Tiempo de caída*: es el tiempo que transcurre desde el final de  $t_s$  hasta que  $I_C=0,1I_{Csat}$ .

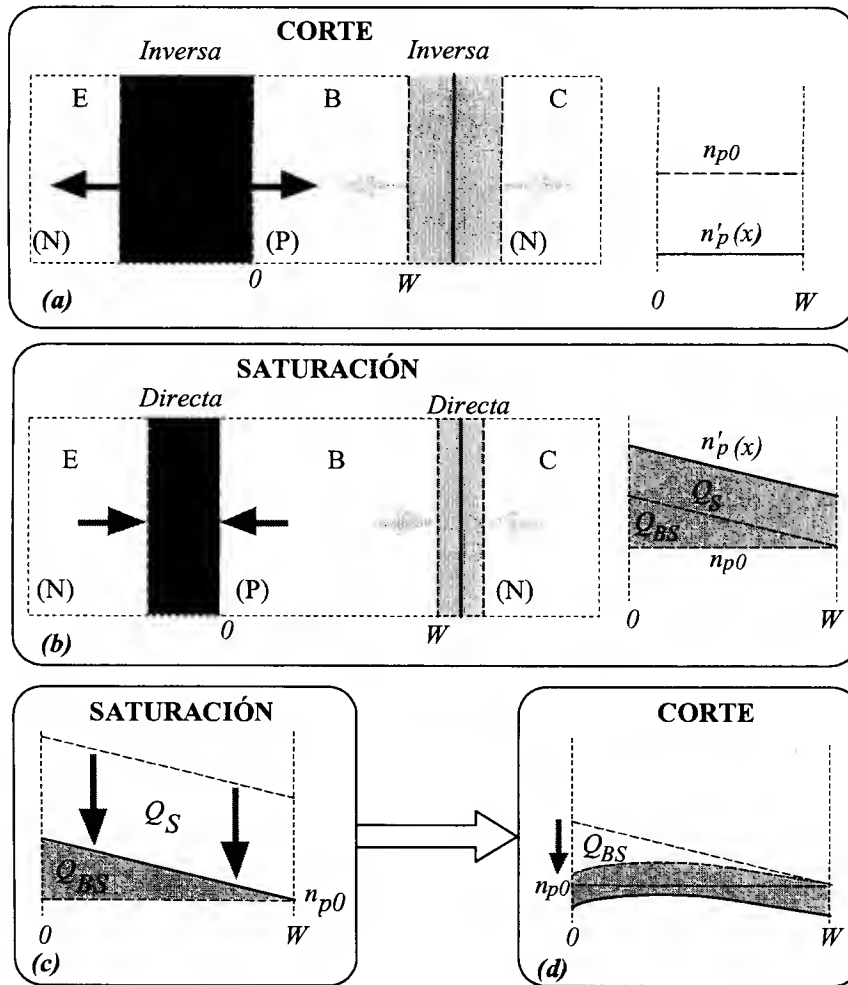
Para  $t=0^-$ ,  $v_i(t) = V_2 < 0$ , y el transistor está en estado de corte, ambas uniones están polarizadas en inversa por lo que las zonas de transición son anchas y en la zona neutra de base existe un defecto de portadores minoritarios, (figura 2.25.a).

En  $t=0$ , aplicamos un pulso a la base. Es decir,  $v_i(t)$  pasa de valer  $V_2 < 0$  a  $V_1 > 0$ , con lo que el transistor pasará a zona activa y si  $V_1$  tiene un valor tal que la unión base-colector pase a estar polarizada en directa, el transistor llegará a la zona de saturación. Cuando el transistor está saturado ambas uniones están polarizadas en directa por lo que las zonas de transición serán más estrechas y en la zona neutra de base existirá un exceso de minoritarios,  $n'_p(x)$ , por encima del valor de equilibrio,  $n_{p0}$  (figura 2.25.b).

Por tanto para que el transistor pase de *corte* a *saturación* se la deberá suministrar a la base la carga suficiente para neutralizar las cargas iónicas existentes en las zonas de transición de ambas uniones y para establecer en la región neutra de base dicho exceso de cargas,  $n'_p(x)$ , suficiente para soportar la demanda de corriente del estado de saturación,  $I_{Csat}$ .

En  $t=t_1$ ,  $v_i(t)$  pasa de valer  $V_1 > 0$  a valer  $V_2 < 0$ , con lo que el transistor pasará del estado de saturación al de corte, y por tanto ahora habrá que extraer el exceso de minoritarios en la zona neutra de base ( $Q_S+Q_{BS}$ ) y las zonas de transición de las uniones se ensancharán de nuevo puesto que pasarán a estar ambas polarizadas en inversa. Mientras se extrae  $Q_S$ , la corriente de colector permanece constante e igual a  $I_{Csat}$ , la recta  $n'_p(x)$  se desplaza hacia abajo paralela a sí misma hasta que  $n'_p(W) = 0$ , que corresponde a  $Q_S = 0$ , figura 2.25.d. Este tiempo desde que se empieza a extraer  $Q_S$  hasta que se anula se le llama *tiempo de almacenamiento* ( $t_s$ ). A partir de este instante el transistor pasa a estar en zona activa y camina hacia zona de corte, para lo cual se deberá extraer  $Q_{BS}$ . Ahora la corriente de colector no permanece constante sino que irá disminuyendo progresivamente (figura

2.25.d). Además, en las zonas de transición habrá una extracción de la carga móvil quedando de nuevo una zona de carga iónica fija más ancha, como corresponde a una unión polarizada en inversa.



**Figura 2.25.** Ilustración del cambio en la zona neutra y en la carga almacenada durante los estados de corte y saturación. (a) Corte. (b) Saturación. (c) Extracción de la carga para salir de saturación. (d) Paso de zona activa hacia el corte.

El tiempo que transcurre desde que se anula  $Q_S$  hasta que desaparece  $Q_{BS}$  y el dispositivo pasa a corte es el *tiempo de caída*. La suma de ambos tiempos, de almacenamiento y de caída, se conoce como *tiempo de corte*,  $(t_s + t_f)$ . Para estos procesos la corriente de base es la encargada de aportar los

portadores necesarios, a la vez que compensar la parte de carga que se pierde por recombinación. Para el análisis de los transitorios es conveniente recordar que el proceso es análogo al que estudiamos en los diodos de unión P-N (apartados 2.4.1 y 2.4.2). De igual forma que allí, aquí tenemos también tres alternativas metodológicas que corresponden a tres grados de aproximación diferentes:

1. *Ecuación de difusión.*
2. *Modelo de control por cargas.*
3. *Circuito equivalente.*

No vamos a desarrollar aquí de forma completa ninguno de los tres métodos porque su complejidad analítica los deja fuera del alcance de este curso. Nos vamos a conformar con el *estudio cualitativo* que acabamos de presentar y con un resumen de los resultados obtenidos al aplicar el modelo de *control por cargas* de forma análoga a como hicimos en el caso de la unión P-N, distinguiendo entre *zona neutra de base* y *zonas de transición*.

**a) Zona neutra.**

Recordemos que los modelos de control por carga estudian los transitorios de conmutación en diodos y transistores a través de la evolución temporal de la carga almacenada en la estructura. Para situaciones estacionarias ( $dQ/dt = 0$ ) podemos reescribir la expresión [2.18] para la zona neutra de base:

$$I_C = \frac{Q_B}{\tau_1} \quad [2.64]$$

$$I_B = \frac{Q_B}{\tau_2} \quad [2.65]$$

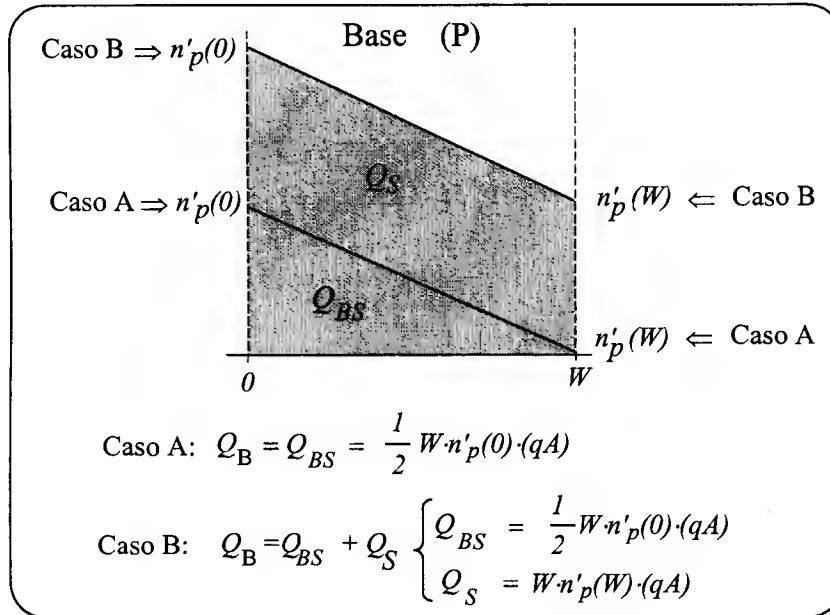
$$I_C = \beta_F I_B, \quad \text{siendo} \quad \beta_F = \frac{\tau_2}{\tau_1} \quad [2.66]$$

Veamos ahora la relación entre la tensión y la carga. El exceso de carga en la zona neutra puede aproximarse por un triángulo de base  $W$  y altura  $n'_p(0)$ , suponiendo que  $n'_p(W)$  es cero ( $Q_B = Q_{BS}$ ). O bien, cuando ambas uniones están muy polarizadas en directa y no podemos despreciar  $n'_p(W)$ , la carga será el área de un rectángulo más la de un triángulo, tal como se ilustra en la *figura 2.26*, ( $Q_B = Q_{BS} + Q_S$ ). En el primer caso tendremos:

$$Q_B = q A \frac{1}{2} W n'_b(0) \quad [2.67]$$

$$\text{Como } n'_b(0) = n_{b0} \left( e^{q V_{BE}/KT} - 1 \right) \quad [2.68]$$

$$\text{tenemos que } Q_B = q \frac{AW}{2} n_{b0} \left( e^{q V_{BE}/KT} - 1 \right) \quad [2.69]$$



**Figura 2.26.** Cálculo sencillo y aproximado de la carga almacenada en la zona neutra de base. En el caso A, la unión de emisor está muy polarizada en sentido directo y la de colector muy poco, por eso despreciamos  $n'_p(W)$ . En el caso B, ambas uniones están muy polarizadas en sentido directo,  $n'_p(W)$  no es despreciable y hay más carga almacenada

Hasta ahora hemos analizado el problema en condiciones estáticas, sin embargo nosotros estamos interesados en las condiciones dinámicas, es decir en conseguir las expresiones de los valores instantáneos de las corrientes en función del tiempo  $[i_c(t), i_b(t)]$ . Para ello vamos a suponer que las velocidades de variación de las tensiones aplicadas entre pares de terminales y de las intensidades de las corrientes son lo suficientemente pequeñas como para que la distribución de carga en la región de base varíe de manera que dé una sucesión de distribuciones estáticas, o sea, consideramos las



*condiciones dinámicas como una sucesión de estados de equilibrio.* Por tanto podemos suponer que la distribución instantánea del exceso de portadores en la zona neutra de base varía de manera que pueda considerarse como una sucesión de distribuciones estacionarias (que seguiremos suponiendo distribuciones triangulares) de forma que la intensidad instantánea de colector es proporcional al valor en ese instante del exceso de carga.

$$i_C = \frac{Q_B(t)}{\tau_I} \quad [2.70]$$

Veamos ahora  $i_B$ . El punto central del razonamiento para poder relacionar la intensidad instantánea de base con el valor en ese instante del exceso de carga en la base es recordar que la base es eléctricamente neutra y que los mayoritarios que se necesitan para neutralizar el exceso de minoritarios sólo los puede suministrar  $i_B$ . Por tanto, la carga de la base  $Q_B$ , sólo estará controlada por  $i_B$  que se encarga de reponer pérdidas por recombinación ( $Q_B/\tau_2$ ) y aumentar o disminuir el exceso ( $dQ_B/dt$ ). La ecuación de conservación de portadores mayoritarios en la base en condiciones dinámicas es entonces,

$$i_B = \frac{dQ_B}{dt} + \frac{Q_B}{\tau_2} \quad [2.71]$$

Finalmente la corriente de emisor será:

$$i_E = i_B + i_C = \frac{dQ_B}{dt} + \frac{Q_B}{\tau_2} + \frac{Q_B}{\tau_I} \quad [2.72]$$

La tensión instantánea emisor-base ( $v_{EB}$ ) estará relacionada con la carga instantánea de igual forma que antes, (ecuación [2.69]) ya que es una consecuencia directa de la naturaleza triangular de la distribución. En resumen, el modelo de control por carga, nos dice que:

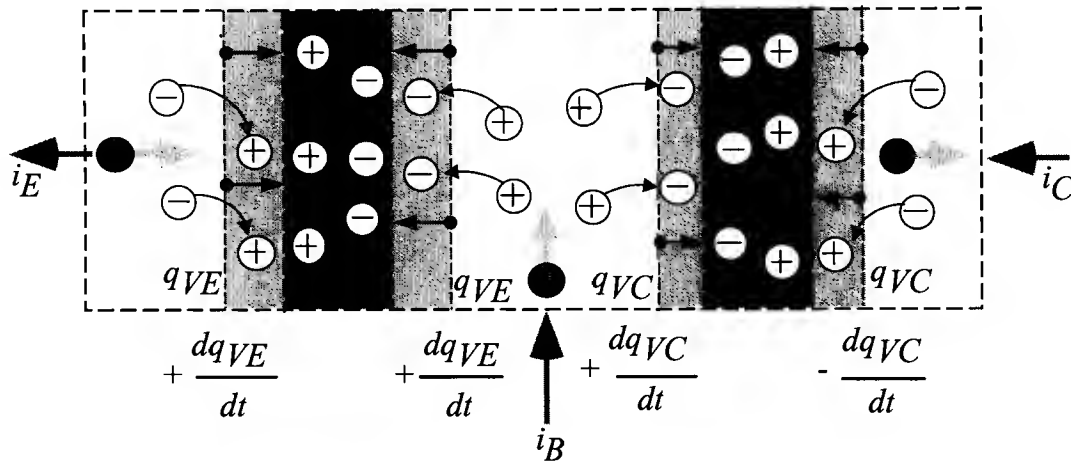
1. La intensidad instantánea en colector,  $i_C(t)$ , está gobernada por el valor instantáneo del exceso de minoritarios en base,  $Q_B$ .
2. Este exceso se neutraliza por otro igual de mayoritarios que está gobernado sólo por  $i_B$ .
3. Las ecuaciones que describen el comportamiento son [2.70], [2.71], [2.72] y [2.69].

**b) Zona de transición.**

Cuando el transistor pasa de corte o saturación a conducción o corte la anchura de las zonas de

transición disminuyen o aumentan, por tanto la corriente de base no sólo debe suministrar portadores para mantener una  $i_C$ , sino que además debe ajustar la carga asociada a las regiones de transición, para dar cuenta de la disminución o aumento de su anchura. El almacenamiento de carga en una y otra mitad de la capa dipolar en la unión de emisor-base es una función de  $v_{BE}$ , que rige la anchura de la capa de carga espacial. Describimos esta carga por  $q_{VE}$ , que es la diferencia entre la carga correspondiente a un valor de  $v_{EB}$  cualquiera y el valor de equilibrio. Así,  $q_{VE} = 0$  para  $v_{EB} = 0$  ( $V_{EB} = cte.$ ), y  $q_{VE} > 0$  para polarización directa, es decir que  $q_{VE}$  es la carga que tiene que aportar la zona neutra de emisor, que coincide con la que tiene que aportar la zona neutra de base al interior de los dos lados de la zona de carga espacial para neutralizarla en parte y hacer que esta disminuya.

De igual forma podemos definir la carga  $q_{VC}$  correspondiente a la unión de colector. Es decir, a un incremento de  $v_{CB}$  le corresponde un incremento de  $q_{VC}$  y por tanto una aportación de portadores mayoritarios hacia el interior de los lados de la zona de transición de la unión de colector (figura 2.27).



**Figura 2.27.** Movimiento de cargas en las zonas de transición.

Por tanto si añadimos estas corrientes de cargas de las zonas de transición a las corrientes asociadas a las zonas neutras tenemos:

$$i_C(t) = \frac{Q_B}{\tau_I} - \frac{dq_{VC}}{dt} \quad [2.73]$$

$$i_B(t) = \frac{Q_B}{\tau_2} + \frac{dQ_B}{dt} + \frac{dq_{VE}}{dt} + \frac{dq_{VC}}{dt} \quad [2.74]$$

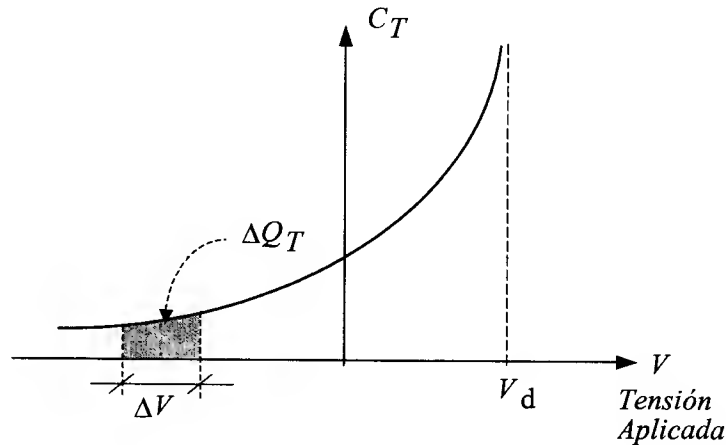
$$i_E(t) = \frac{Q_B}{\tau_1} + \frac{Q_B}{\tau_2} + \frac{dQ_B}{dt} + \frac{dq_{VE}}{dt} \quad [2.75]$$

A veces nos puede interesar expresar  $dq_{VE}/dt$  y  $dq_{VC}/dt$  en función de las capacidades de transición, para ello como la variación de estas cargas es un proceso no lineal lo tratamos en términos de una capacidad media. La capacidad de transición varía con la tensión aplicada de la forma que se muestra en la *figura 2.28*.

Así podemos poner:

$$i_c(t) = \frac{Q_B}{\tau_1} - C_{TC} \frac{dv_{CB}}{dt} \quad [2.76]$$

$$i_B(t) = \frac{Q_B}{\tau_2} + \frac{dQ_B}{dt} + C_{TE} \frac{dv_{EB}}{dt} + C_{TC} \frac{dv_{CB}}{dt} \quad [2.77]$$



**Figura 2.28.** Variación de la capacidad de transición con el potencial.

El siguiente paso sería representar el transistor por un circuito equivalente que ponga de manifiesto su comportamiento en conmutación según este modelo de control por cargas, es decir tendríamos que obtener un circuito que verificase las ecuaciones [2.73], [2.74] y [2.75] y sobre él

calcular los tiempos de subida, retardo, almacenamiento y bajada. Su desarrollo queda de nuevo fuera de la extensión que queremos darle a este apartado y puede encontrarse, entre otros muchos sitios, en el texto "Electrónica Analógica Lineal" editado por la UNED.

## 2.7. ESTRUCTURA MIS

Hasta ahora hemos estudiado sólo dispositivos bipolares (diodos de unión P-N y transistores bipolares), llamados así porque en el transporte controlado de carga intervenían los dos tipos de portadores, electrones y huecos. Una parte muy importante de la Electrónica Digital se basa sin embargo en los transistores de efecto campo cuyo principio de funcionamiento es el control por un campo externo (a través de un terminal de puerta, G) de la conductividad de un canal de carga móvil que enlaza dos electrodos externos (fuente, S y drenador, D) con la polarización adecuada.

Hay esencialmente dos tipos de transistores de efecto campo (FET)

✧ JFET (FET de unión P-N)

✧ MOSFET (FET Metal-Oxido-Semiconductor)

El JFET fue ideado por Shockley en 1952 y realizado por Dacey y Ross en 1955 y no será estudiado aquí porque está dirigido esencialmente al campo de la Electrónica Analógica. A pesar de ser unipolar, el JFET es un dispositivo "de volumen", es decir los procesos importantes ocurren en el volumen del semiconductor ya que el canal se crea mediante dos uniones P-N polarizadas en inversa.

En cambio, el transistor MOS es un dispositivo de superficie, en el que la conducción se realiza en un canal muy próximo a la superficie del semiconductor. Hasta ahora siempre habíamos estudiado a los semiconductores como cristales perfectos e infinitos. Las únicas imperfecciones eran los átomos de elementos de los grupos III (Boro, Aluminio) y del grupo V (Fósforo,...) que usábamos en pequeñísima proporción para impurificar al semiconductor intrínseco (Silicio, Arseniuro de Galio) y pasarlo a tipo P (con conducción dominada por huecos) ó a tipo N (con conducción dominada por electrones). Ahora en cambio vamos a estudiar un dispositivo asociado al carácter finito del cristal que donde acaba (su superficie) rompe la periodicidad de la red cristalina. Esa ruptura de la periodicidad introduce nuevos estados electrónicos permitidos y con ellos, al ionizarse, una barrera de potencial. Veremos enseguida que esa barrera da origen a las estructuras MIS (Metal-Insulator-Semiconductor) y a los transistores MOS (Metal-Óxido-Semiconductor) que son una yuxtaposición de

estructuras MIS.

Es un hecho experimental que en la superficie de un semiconductor real aparece una barrera de potencial asociada a cierta distribución de carga espacial inmóvil. Esta barrera puede ser producida por:

- 1) Aplicación de un campo externo.
- 2) Contacto con otro material de distinto potencial de contacto.
- 3) Existencia de estados superficiales con energías,  $E_t$ , en la región que antes era de energías prohibidas, debidos a la ruptura de la periodicidad del cristal o a impurezas. De las tres causas posibles, Bardeen demostró que la dominante era la tercera. Estos estados superficiales pueden tener una energía  $E_t$ , próxima a la banda de valencia ( $E_v$ ) y serán por tanto estados aceptores. En cambio, si  $E_t$  está próximo a la banda de conducción ( $E_c$ ), serán estados dadores.

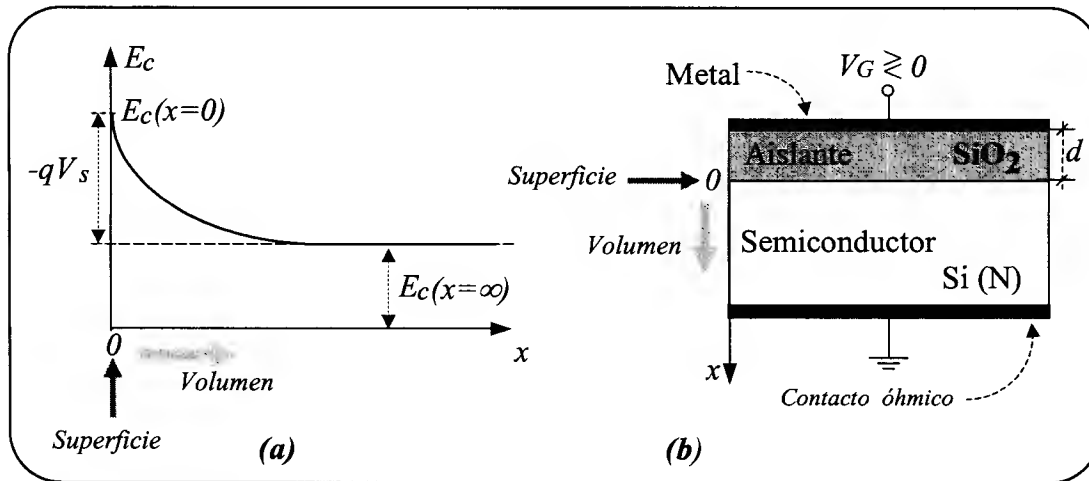
Decir que un estado superficial es dador o aceptor es fijar el signo de la carga iónica que va a generar y, por consiguiente, el signo de la barrera de potencial en la superficie en relación con el potencial en el volumen. Así, si pudiéramos realizar un experimento ficticio consistente en introducir un conjunto de estados neutros, a temperatura ambiente se ionizarían rápidamente cargándose negativamente los aceptores (han aceptado un electrón) y positivamente los dadores (han dado un electrón y quedan con una carga positiva en exceso).

Sea cual fuere la contribución de cada una de las tres causas, en una estructura MIS aparece el efecto de su superposición como una carga neta inmóvil en la superficie que produce una barrera de potencial, tal como la que se ilustra en la *figura 2.29.a*. La estructura MIS se muestra en la *figura 2.29.b* donde  $d$  es el ancho de la capa de aislante ( $\text{SiO}_2$ , por ejemplo) y  $V$  es el potencial aplicado, positivo cuando el metal queda polarizado positivamente respecto al contacto óhmico y negativo en caso contrario.

Supongamos que inicialmente el potencial en la superficie es nulo,  $V_s = 0$  y apliquemos una tensión de polarización externa positiva o negativa. Podemos encontrarnos entonces con alguna de las tres situaciones de la *figura 2.30*, (a) *Acumulación*, (b) *Vaciamiento* y (c) *Inversión*.

**Acumulación.** Decimos que tenemos una situación de acumulación en la superficie de un semiconductor tipo N cuando la concentración de electrones es mayor en la superficie que en el volumen. Esta situación puede estar producida por la aplicación de un campo externo,  $V_G > 0$ , ó por

la existencia de estados superficiales dadores. En ambos casos, el aspecto de la barrera de potencial en las proximidades de la superficie es el de la *figura 2.30.a*.

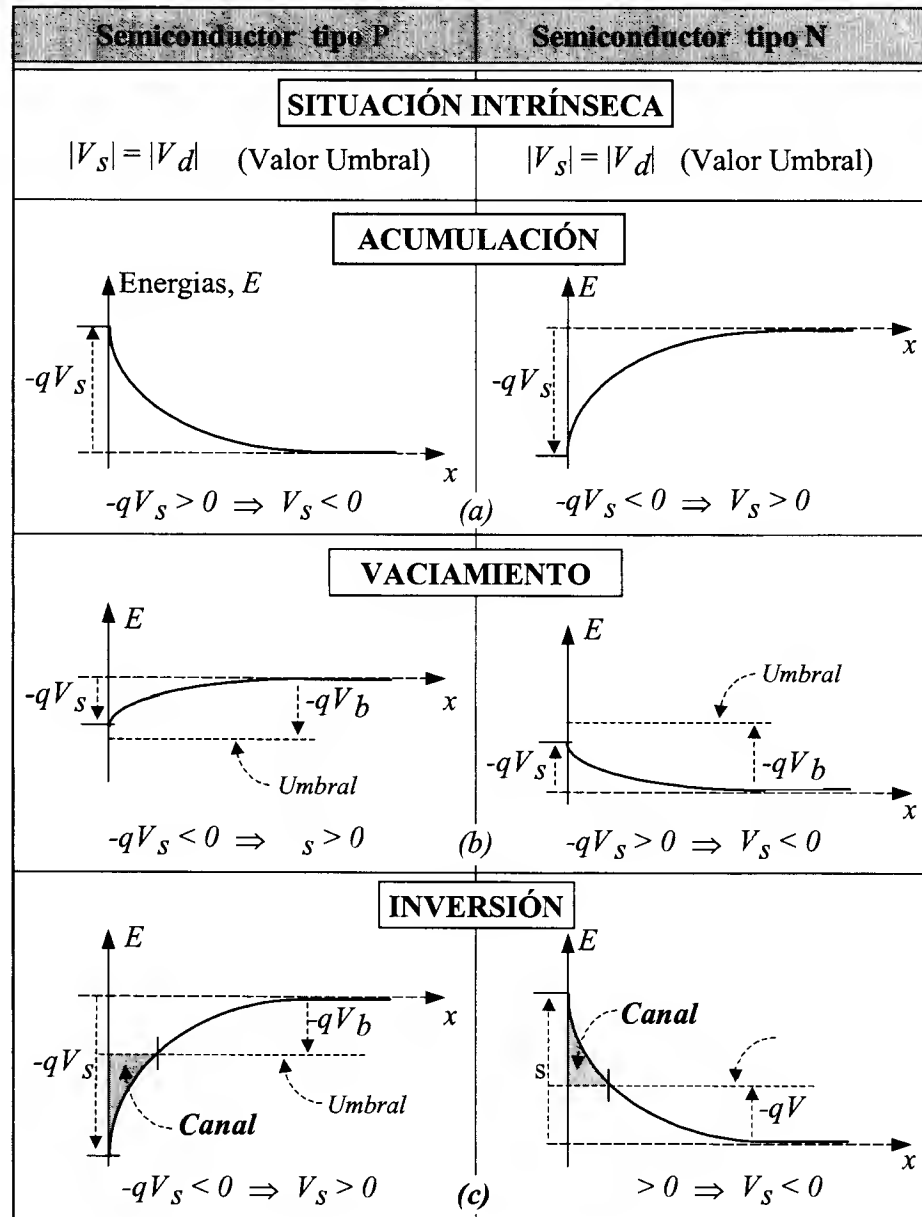


*figura. 2.29.* Barrera de potencial en una estructura MIS. (a) Barrera. (b) Estructura MIS sobre un semiconductor tipo N.

**Vaciamiento e inversión.** Inversamente al caso anterior decimos que existe una situación de vaciamiento o inversión cuando la concentración de portadores mayoritarios en las proximidades de la superficie es inferior a la del volumen. Lógicamente, la concentración de minoritarios aumentará para mantener la neutralidad eléctrica.

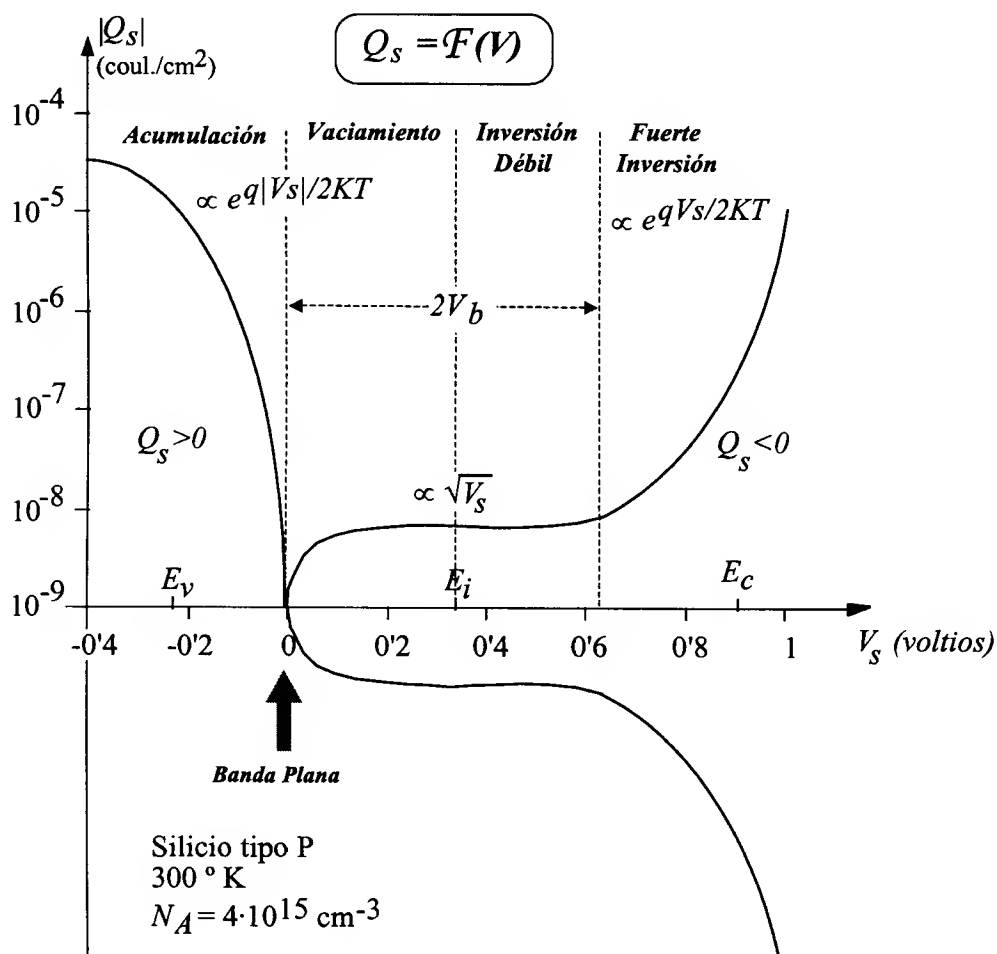
Esta situación puede estar producida por la aplicación de un campo externo ( $V < 0$ ), por la presencia de estados superficiales aceptores o por una combinación de ambos efectos. Las *figuras 2.30.b* y *c* muestran la evolución del potencial en las proximidades de la superficie. En *2.30.b* se muestra una situación de vaciamiento, pero si seguimos aumentando el campo externo puede presentarse una situación de inversión, llamada así porque en la superficie aparecen como mayoritarios aquellos portadores que en el volumen son minoritarios. Esto ocurre para un cierto valor de  $V$  (negativo para semiconductores tipo N y positivo para tipo P).

En la *figura 2.30* vemos como las condiciones de acumulación, vaciamiento e inversión para semiconductores tipo P y N sólo dependen del valor del potencial en la superficie,  $V(x=0) = V_s$ .



**Figura 2.30.** Condiciones de acumulación, vaciamiento e inversión en estructuras MIS sobre semiconductores tipo P y N. El potencial,  $V_s$ , lo marca la tensión externa,  $V_G$ . (a) Acumulación. (b) Vaciamiento. (c) Inversión.

La figura 2.31 muestra la relación entre el potencial en la superficie que da lugar a las situaciones de acumulación, vaciamiento o inversión y el valor de la carga tanto en la región de carga espacial iónica (inmóvil) como en el canal de carga móvil. Su obtención a partir de la ecuación de Poisson queda fuera del alcance de este curso. Su significado es, sin embargo, accesible sin la analítica que lo soporta. Los datos proceden de un semiconductor tipo P impurificado con una densidad de átomos aceptores,  $N_A = 4 \cdot 10^{15} \text{ cm}^{-3}$  y operando a temperatura ambiente.  $V_s$  es el potencial en la superficie,  $V_b$  el valor umbral de referencia que marca el potencial que existiría en un semiconductor intrínseco,  $Q_s$  es la carga y  $\mathcal{F}$  la función que la relaciona con el potencial.



**Figura 2.31.** Variación de la carga con el potencial mostrando cómo se pasa desde acumulación a inversión.



Para  $V_s < 0$ , la carga es positiva y estamos en condiciones de acumulación. Aquí  $F$  puede aproximarse por una exponencial decreciente. Al seguir recorriendo la curva de izquierda a derecha encontramos la condición de banda plana,  $V_s = 0$  y  $Q_s = 0$ . Posteriormente,  $0 < V_s < V_b$ , entramos en la zona de vaciamiento con  $Q_s < 0$ . Finalmente para  $V_s \gg V_b$  entramos en zona de inversión en la que la función  $F$  está dominada por un término exponencial.

Suele considerarse, por elegir un valor cómodo, que la zona de fuerte inversión comienza cuando el potencial en la superficie es el doble del umbral de referencia

$$V_s \text{ inv} \approx 2V_b \quad [2.78]$$

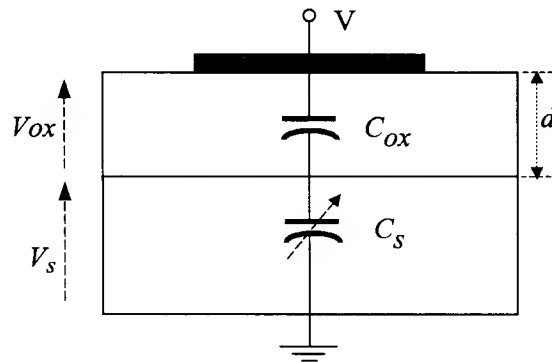
Como  $V_b$  depende de la concentración de impurezas, a través de esta concentración tenemos el control de las condiciones de inversión en un semiconductor extrínseco.

Un amplio rango de aplicaciones en electrónica digital de los transistores MOS y las estructuras MIS que les dan origen es consecuencia de el alto grado de aislamiento en torno a una capacidad de forma tal que podemos almacenar un paquete de carga de forma controlada. Dependiendo del valor de este paquete de carga diremos que la estructura almacena un cero o un uno. Esta función de memoria con dos estados distinguibles mantenidos de forma casi permanente es físicamente posible gracias al aislamiento inherente a las estructuras MIS que les permite almacenar carga en la superficie ( $Q_s$ ) y en el óxido ( $Q_{ox}$ ), de forma que un trocito de una estructura MIS puede considerarse eléctricamente como la combinación en serie de dos capacidades,  $C_s$  y  $C_{ox}$ , tal como se ilustra en la *figura 2.32*, dando lugar a una capacidad total,

$$C = \frac{C_{ox} \cdot C_s}{C_{ox} + C_s} \quad [2.79]$$

Veremos después al hablar de estructuras FAMOS en el tema 7 la importancia que tiene esta expresión para acomodar los procesos de lectura y escritura en una celda de memoria mediante procedimientos eléctricos (memorias EEPROM)

La expresión de  $C_s$  puede obtenerse a partir de  $Q_s$  y la del óxido es simplemente,  $C_{ox} = \epsilon_{ox}/d$ , siendo  $\epsilon_{ox}$  la constante dieléctrica y  $d$  la profundidad de la capa. No nos interesa sin embargo el desarrollo analítico sino el hecho conceptual de la aparición de esta capacidad en la estructura MIS.



**Figura 2.32.** Capacidad de la estructura MIS.

## 2.8. TRANSISTORES MOSFET.

Se les llama transistores de puerta aislada (Insulated Gate Field Effect Transistor, IGFET) ó MOS (haciendo referencia a su estructura física: Metal-Óxido-Semiconductor). El efecto amplificador del MOS está basado en el control de la conductividad de un canal superficial (control de su densidad de carga móvil), que conecta los terminales de drenador y fuente mediante el campo eléctrico transversal creado por el potencial del terminal de puerta. Este terminal está aislado eléctricamente de la fuente y del drenador por una capa de  $\text{SiO}_2$ .

En función de la polaridad del canal hay dos tipos de MOS: de canal N y de canal P. En cada caso, según el modo de operación, podemos distinguir dos tipos:

MOS de vaciamiento (depletion)

MOS de realce (enhancement)

Este dispositivo es especialmente importante debido a la sencillez de su estructura física (lo que permite una alta densidad de empaquetamiento para integración en gran escala LSI y VLSI), a su alta impedancia de entrada, bajo consumo (en especial la familia C-MOS) y a su facilidad de aislamiento. Los microprocesadores, las memorias EPROM, RAM, FIFO y CAM y toda la electrónica digital de alta densidad están dominados por la tecnología MOS.

Consideremos primero las características del MOSFET tipo P (realce) y canal largo. Es decir, con  $L \gg (W_S + W_D)$ , siendo L la longitud del canal y W los anchos de las zonas de transición de las

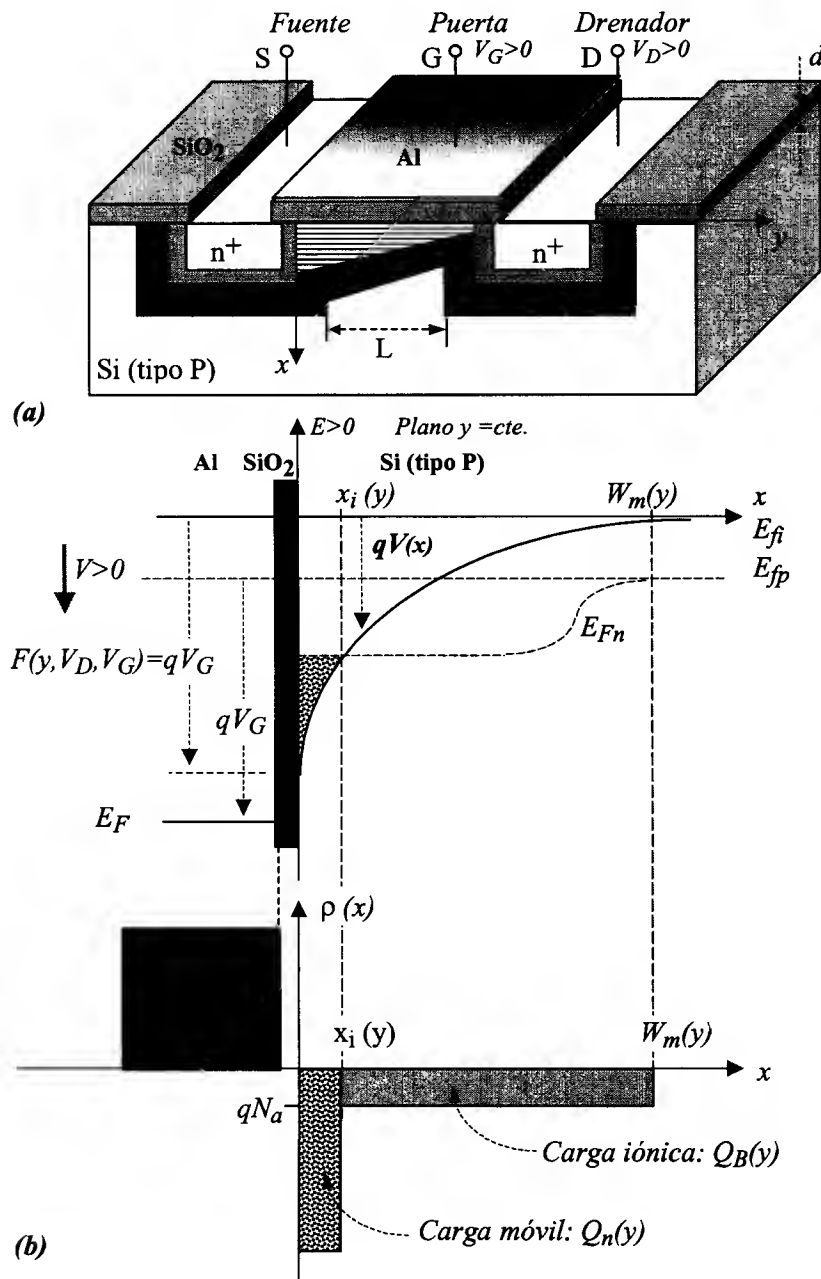
uniones de drenador y fuente. Esto nos servirá de base para el estudio de los MOSFET de canal corto,  $L \leq (W_S + W_D)$  y otros dispositivos análogos necesarios para pasar la frontera de la VLSI.

Un transistor MOS se puede considerar como una yuxtaposición de diodos MIS cuyos parámetros geométricos y tensión de polarización varían gradualmente entre los terminales de fuente y drenador. La *figura 2.33* ilustra la estructura física ideal de un MOSFET de canal N. Se parte de un sustrato de silicio tipo P en el que por los métodos clásicos de la tecnología planar (máscaras, ventanas, difusión, etc...) se hacen crecer dos regiones  $N^+$  que constituirán los terminales de drenador y fuente. Posteriormente, sobre la capa de óxido de silicio ( $SiO_2$ ) se deposita una capa metálica de aluminio o silicio policristalino que formará el electrodo de puerta.

Vamos a describir el funcionamiento del dispositivo en modo de realce, es decir, cuando no hay polarización en la puerta ( $V_{GS} = 0$ ) no hay canal, de manera que partimos de conductividad nula entre drenador y fuente ( $I_D = 0$ ). La única corriente que puede pasar en estas condiciones, si la tensión de drenador es elevada es la corriente inversa de uno de los diodo  $N^+P$  que aparecen en oposición entre drenador y fuente.

Al aplicar un potencial al terminal de puerta ( $V_G > 0$ ) e ir aumentándolo progresivamente, para un cierto valor ( $V_G > V_T$ ) se crea un canal de inversión entre las dos regiones  $N^+$ : así drenador y fuente quedan conectados por un canal superficial, tipo N, de alta conductividad (canal de inversión) cuyo valor aumenta con el potencial de puerta hasta un cierto valor máximo en condiciones de saturación. Tras crearse la situación de inversión, la capa iónica y el canal protegen al volumen semiconductor de posteriores penetraciones del campo (efecto "espejo"). Los posteriores incrementos de la tensión de puerta se absorben en la capa de inversión, aumentando la densidad de carga libre.

Veamos ahora cualitativamente cómo influye la tensión de drenador,  $V_D$ , en cada "loncha" de estructura MIS situada en la coordenada  $y$ , entre la fuente ( $y=0$ ) y el drenador ( $y=L$ ). Al aplicar un potencial positivo, ( $V_G > 0$ ), en el terminal de puerta se repiten los mismos fenómenos estudiados en el diodo MIS, pasando por situaciones sucesivas de vaciamiento, inversión y fuerte inversión, a medida que aumentamos el valor de  $V_G$ . Siempre que  $V_G$  supere a  $V_T$ , es decir, siempre que la tensión de puerta sea suficiente para (descontadas las caídas en el óxido, estados superficiales y diferencias de potencial de contacto) que quede sobre la superficie un potencial mayor o igual a  $2V_b$  tendremos inversión. Esto es lo que ocurría en la estructura MIS y lo que ocurre en el terminal de fuente del MOS.



**Figura 2.33.** Transistor MOS de canal N. (a) Estructura física. (b) Forma del potencial en las proximidades de la superficie para un plano  $y = \text{cte.}$  situado en un punto intermedio entre fuente (S) y drenador (D). Se muestra también la distribución de carga en la zona de transición y en la capa de inversión.

Ahora bien, al aplicar un potencial positivo en el drenador,  $V_D > 0$ , hacen falta valores mayores de  $V_G$  para producir inversión. El potencial necesario para producir inversión ha aumentado y es función de la posición en el canal.

$$V_s(\text{inv})|_y = 2 \cdot V_b + V(y), \quad \text{con } V(0) = 0 \text{ y } V(L) = V_D \quad [2.80]$$

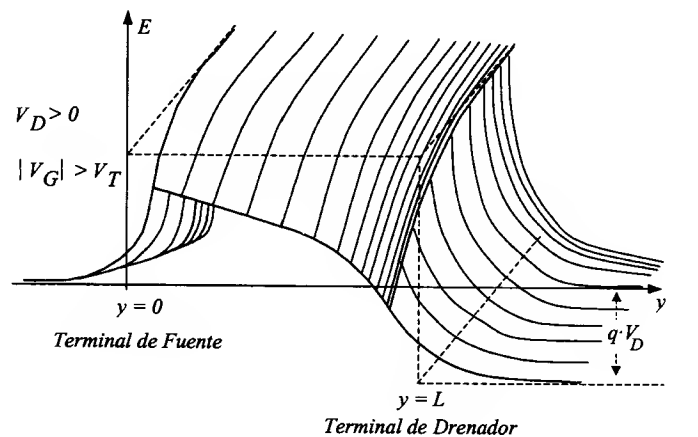
Es decir, para producir inversión cerca de la fuente basta con  $V_s = 2V_b$  (o lo que es equivalente,  $V_G \geq V_T = Q_s/C_{ox}$ ). En cambio, para producir inversión en el terminal de drenador necesitamos un potencial en la superficie,

$$V_s(L) = 2V_b + V_D \quad [2.81]$$

o lo que es equivalente

$$V_G \geq V_T = \frac{Q_B}{C_{ox}} + 2V_b + V_D \quad [2.82]$$

El canal se crea antes en la unión de fuente que en la de drenador. Inversamente, si una vez creado el canal se aumenta  $V_D$  ó se disminuye  $V_G$ , el canal empezará a estrangularse antes en las proximidades del drenador, donde  $V(y)$  es máxima e igual a  $V_D$ . La *figura 2.34* muestra un resumen gráfico del comportamiento estático del transistor MOS. En vez de dibujar las curvas del potencial hemos dibujado las correspondientes a la energía ( $E = q \cdot V$ )



**Figura 2.34.** Evolución del potencial,  $V(y)$ , y la energía,  $E = q \cdot V(y)$ , a lo largo del canal entre fuente y drenador.

## 2.9 CURVAS CARACTERÍSTICAS, REGIONES DE FUNCIONAMIENTO Y MODELO ELEMENTAL DEL TRANSISTOR MOS-FET

Hemos visto el principio de funcionamiento del transistor MOS. El siguiente paso es obtener sus curvas características,  $I_D = f(V_D)$ , para distintos valores de la tensión de puerta,  $V_G$ . Al aplicar una tensión de puerta  $V_G = cte.$  y suficientemente grande como para crear el canal de inversión, la corriente en el canal,  $I_D$ , será función de  $V_D$  y su expresión analítica se obtiene integrando la densidad de corriente sobre una sección transversal del canal. Su valor es,

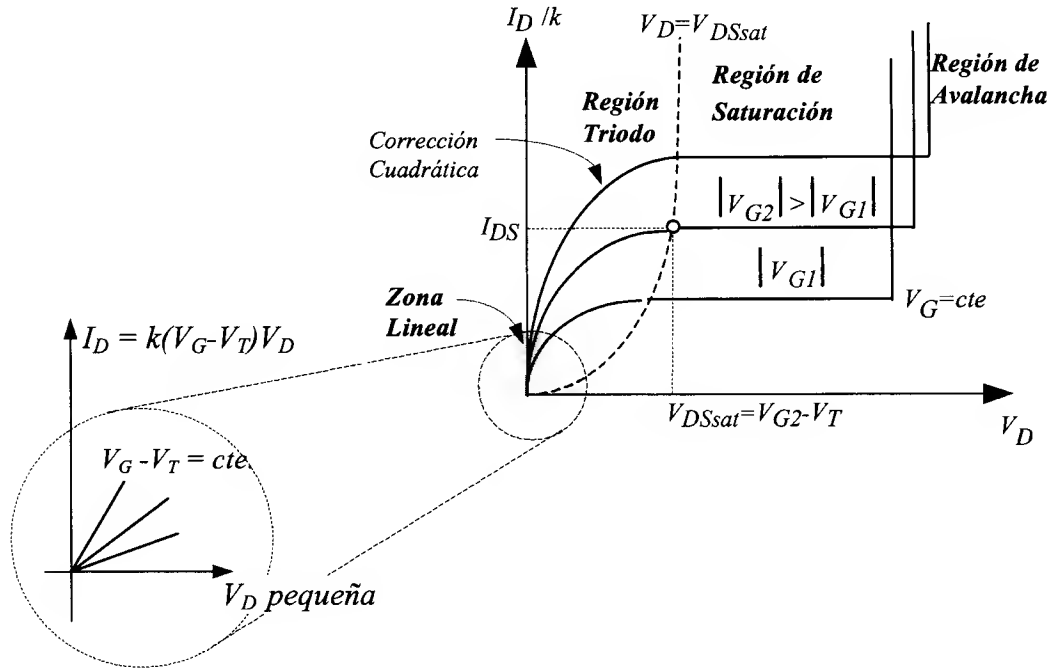
$$I_D = k \left[ (V_G - V_T) \cdot V_D - \frac{V_D^2}{2} \right] \quad [2.83]$$

siendo  $V_T$  la tensión umbral y  $k$  una constante función de la geometría del dispositivo y de las constantes del semiconductor.

Veamos ahora las distintas regiones de funcionamiento. La *figura 2.35.a* muestra las curvas características con una ampliación de la zona lineal. Al aplicar un potencial en el terminal de puerta,  $V_G$ , mayor que el umbral,  $V_T$ , se crea el canal de inversión que conecta drenador y fuente. Si entonces aplicamos un potencial en el drenador positivo respecto de la fuente y pequeño ( $V_D \ll V_G - V_T$ ) aparece una corriente de drenador,  $I_D$ , que es proporcional a  $V_D$  para  $V_G$  constante. Estamos en la región lineal o triodo en la que el canal se comporta como una resistencia cuyo valor se puede controlar mediante  $V_G$ . Analíticamente podemos aproximar la expresión [2.83] despreciando la corrección cuadrática,  $\frac{1}{2}V_D^2$ , así:

$$\left. \begin{array}{l} (V_G - V_T) > 0 \\ V_D > 0 \\ V_D \ll (V_G - V_T) \end{array} \right\} \Rightarrow I_D \approx k (V_G - V_T) V_D \quad [2.84]$$

Al aumentar  $V_D$ , conservándose todavía inferior a  $(V_G - V_T)$ , actúa la corrección cuadrática y entramos en la parte curva de la región triodo, para la que no es válida la aproximación lineal y debemos usar la expresión completa, [2.83]. Para cada  $V_G$  y  $V_T$  existe una *tensión de saturación*,  $V_{DSsat} = V_G - V_T$ , que marca el límite de la zona.



**Figura 2.35.** Curvas características y regiones de funcionamiento del transistor MOS.

Si aumentamos  $V_D$  hasta el valor  $V_{DSsat}$ , se estrangula el canal en  $y = L$  y comienza la región de saturación. A partir de aquí, posteriores incrementos en  $V_D$ , ( $V_D > V_{DSsat}$ ), dejan constante a la corriente del drenador, que mantiene el valor máximo de la región anterior.

$$\left. \frac{\partial I_D}{\partial V_D} \right|_{V_G=cte} = 0 \Rightarrow k[(V_G - V_T) - V_{DSsat}] = 0 \Rightarrow V_{DSsat} = V_G - V_T \quad [2.85]$$

$$I_{Dsat} = \frac{k}{2}(V_G - V_T)^2 = \frac{k}{2}V_{DSsat}^2 \quad [2.86]$$

Posteriores incrementos en  $V_D$ , acortan el canal acercando el punto de estrangulamiento hacia la unión de fuente. Si el estrangulamiento ocurre en  $y=L'$  (con  $L' < L$ ) allí  $V = V_{DSsat}$ , el resto,  $V_D - V_{DSsat}$ , cae en la zona de transición de la unión P-N<sup>+</sup> de drenador. Podemos razonar acerca de la constancia de la *corriente de saturación*,  $I_{Dsat}$ , pensando que el número de portadores móviles que llegan a  $L'$  es prácticamente el mismo que llegaba a  $L$ , y que de  $L'$  a  $L$  caminan sin recombinación apreciable, a velocidad límite, en el fuerte campo ( $\approx 105$  voltios/cm), de la zona de transición por un

mecanismo análogo al de recogida de portadores en la unión de colector de un transistor bipolar trabajando en modo activo directo.

El comportamiento del MOSFET para pequeñas señales se puede representar por un amplificador a transconductancia bastante próximo al ideal, según la *figura 2.36.b*, cuyos parámetros característicos dependen de la posición del punto de funcionamiento estático. Para establecer un punto de funcionamiento (punto  $Q$ ), tenemos que polarizar al transistor MOS de forma análoga a como polarizábamos los transistores bipolares. La polaridad de las fuentes externas depende del tipo de MOS que estemos usando. La *figura 2.36.a* muestra el esquema de polarización en fuente común de un MOS de realce (canal N) y dos rectas de carga que generan puntos estáticos en la zona lineal ( $Q_L$ ) y en la recta de saturación ( $Q_S$ ).

Veamos ahora el cálculo de la conductancia de drenador y la transconductancia en estos puntos:

$$g_d = \left. \frac{\partial I_D}{\partial V_D} \right|_{V_G} = k[(V_G - V_T) - V_D] \quad [2.87]$$

Vemos pues que la conductancia de drenador,  $g_d$ , crece con  $V_G - V_T$  y decrece linealmente con  $V_D$  anulándose para  $V_D = V_{DSsat}$  (final de la zona triodo).

La transconductancia es:

$$g_m = \left. \frac{\partial I_D}{\partial V_G} \right|_{V_D} = k \cdot V_D \quad [2.88]$$

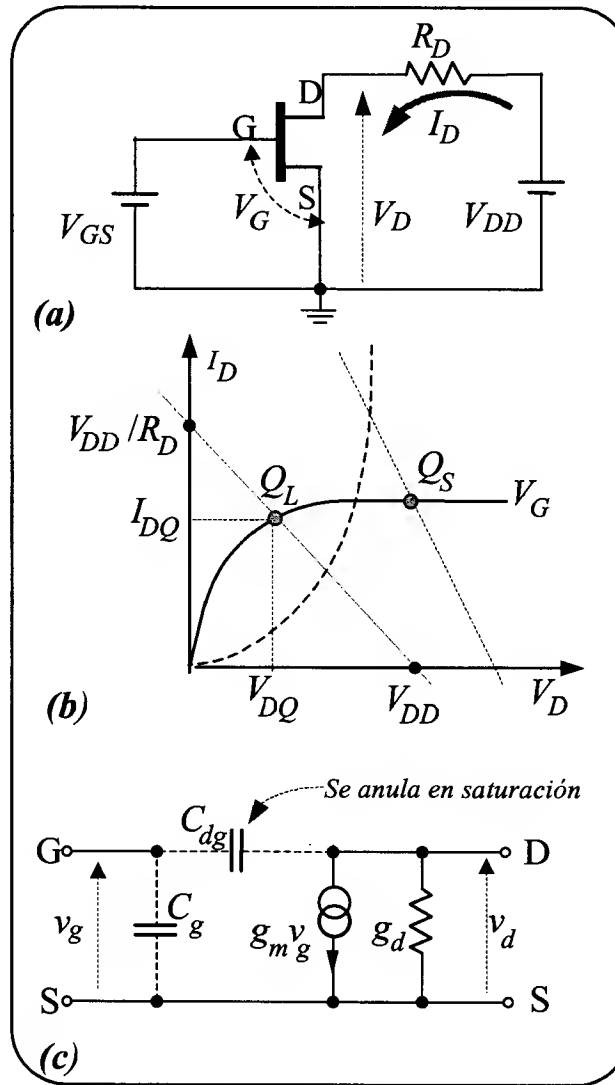
Como vemos  $g_m$  es independiente de la tensión de puerta y aumenta con  $V_D$  hasta la zona de saturación en donde permanece constante e igual al valor máximo de la zona anterior.

La ganancia en tensión del dispositivo es

$$\mu = - \left. \frac{\partial V_D}{\partial V_G} \right|_{I_D} = \frac{g_m}{g_d} = \frac{V_D}{V_G - V_T - V_D} \quad [2.89]$$

En el modelo ideal esta ganancia se haría infinita al entrar en zona de saturación ( $g_d=0$ ) pero las consideraciones sobre el acortamiento del canal y otros efectos secundarios del dispositivo real justifican un valor finito para  $g_d$ , y por consiguiente para  $\mu$ .

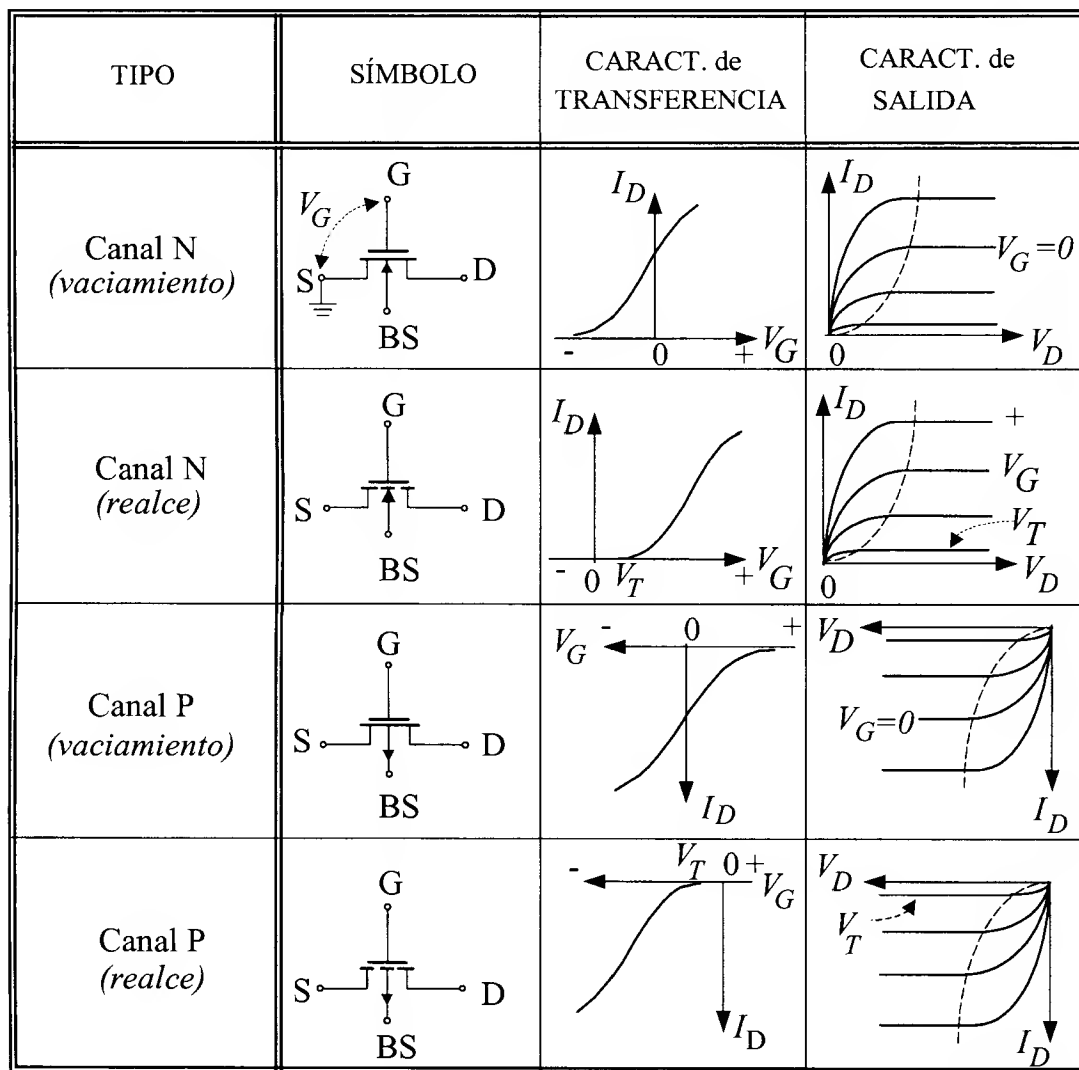




**Figura 2.36.** (a) Esquema de polarización de un transistor MOS de cana N y modo de realce. (b) Ilustración de dos puntos de trabajo,  $Q_L$  y  $Q_S$ , sobre una característica de salida definida por un valor constante en la tensión de puerta  $V_G$ . Obsérvese que esta tensión es en realidad la diferencia de potencial entre los terminales de puerta y fuente, sólo que aquí el terminal de fuente está a tierra. (c) Circuito equivalente para pequeñas señales. Es decir, para describir el comportamiento lineal en torno a los puntos de funcionamiento estático ( $Q_L$  y  $Q_S$ ). Se trata de una versión simplificada en la que la impedancia de entrada posee una componente resistiva tan alta que, esencialmente, queda caracterizada por el valor de una capacidad,  $C_g$ .

Estos son parámetros "intrínsecos" que no incluyen las capacidades de difusión y las de transición drenador-sustrato, etc... No obstante, desde el punto de vista práctico nos basta con completar el modelo incluyendo las capacidades (línea de puntos) de puerta y drenador ( $C_g$ ,  $C_{dg}$ ).

La *figura 2.37* muestra las características de salida y de transferencia de los distintos tipos de transistores MOS. Obsérvese que se pueden pasar de unas a otras mediante traslaciones y giros.



**Figura 2.37.** Resumen de los distintos tipos de transistores MOS.

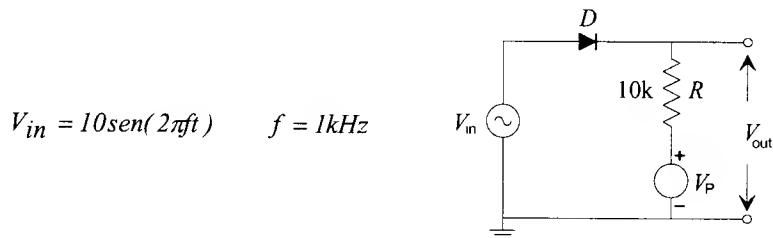
Durante todo este apartado hemos estudiado un tipo concreto de transistor MOS: canal N y trabajando en modo de realce. Sin embargo, tal como se ilustra en la figura 2.37, hay 4 tipos de transistores MOS. Para canal N, podemos hacerlo trabajar en modo de realce o en modo de vaciamiento. Lo mismo podemos hacer para el caso de canal P. En el modo de realce para  $V_G = 0$  no hay canal, lo creamos al aplicar el potencial de puerta  $V_G$ . Si el canal es N, la tensión es positiva ( $V_G > V_T$  y  $V_T > 0$ ). Si el canal es P,  $V_T$  es negativa. En el modo de vaciamiento, antes de aplicar  $V_G$  ya hay canal, de forma que las características de transferencia ocupan dos cuadrantes. Estos dispositivos admiten potenciales positivos y negativos para el terminal de puerta.

## 2.10. PROBLEMAS (\*)

**E.2.1.** Con ayuda del circuito básico de polarización de un diodo, obtener la curva característica del mismo por simulación. ¿Qué sucede cuando excitamos con onda senoidal de la forma:  $V_{in} = 5 \sin(2\pi ft)$ ? Obtened por simulación la forma de onda de la señal de tensión en bornes de la resistencia y del diodo y explicar la relación entre ambas. Explíquense detalladamente las curvas obtenidas. Móntese el circuito con un diodo real y obténganse experimentalmente los valores y curvas pedidos.

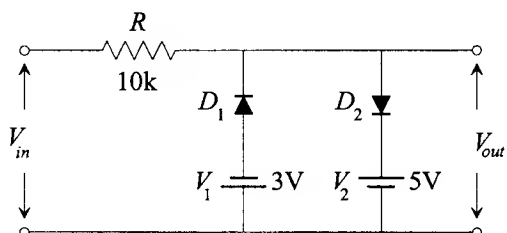
**E.2.2.** Dado el circuito de la figura adjunta:

- Obtener su característica de transferencia para los casos  $V_p = 5V$  y  $V_p = -5V$ .
- Dibujar la forma de la señal de salida en ambos casos con excitación sinusoidal.
- Tomando  $R = 50\Omega$  calcular el valor de la resistencia del diodo en conducción a partir de los datos de la simulación.

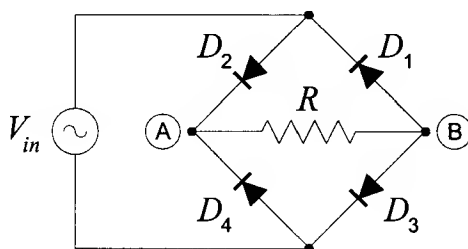


(\*) La solución de estos ejercicios se encuentra en el texto de problemas.

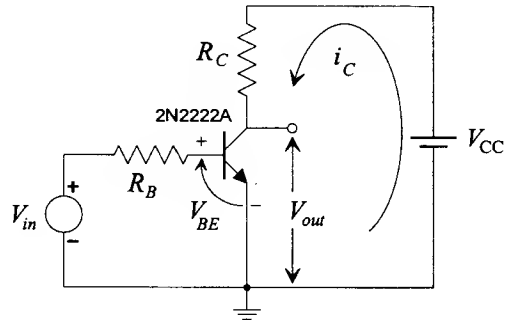
- E.2.3.** Obtener la característica de transferencia del circuito de la figura. Dibujar la señal de salida cuando a la entrada tenemos la señal:  $V_{in} = 10 \sin(2\pi ft)$  con  $f=1\text{kHz}$ .



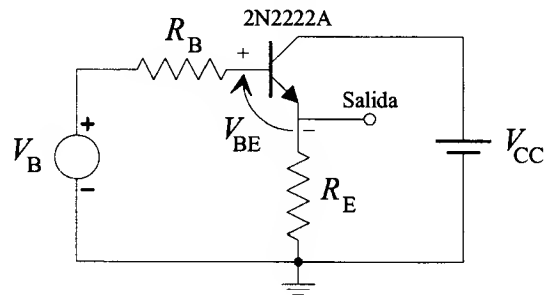
- E.2.4.** Dado el circuito rectificador de la figura adjunta, obtener la forma de onda de la señal de tensión sobre la resistencia  $R$  ( $V_{AB}$ ) cuando excitamos con una señal sinusoidal de 10 Voltios de amplitud. Realizar la simulación del circuito y comprobar que convierte la señal sinusoidal de entrada en una señal rectificada en doble onda siempre positiva.



- E.2.5.** Diseñar una experiencia que permita obtener la característica de entrada de un transistor bipolar y la familia de características de salida en configuración de emisor común. Comprobar sobre el circuito obtenido que la unión Base-Emisor se comporta como un diodo. Obtener las curvas características mencionadas por simulación y de forma experimental para un transistor comercial. Supóngase  $V_{CE} = 5\text{V}$ .
- E.2.6.** Usar las curvas características del problema anterior para polarizar un transistor NPN de forma que trabaje únicamente en zona activa, de acuerdo con el circuito de polarización que se muestra en la figura. Es decir, no entre en corte ni en saturación, cuando se excita con el generador:  $V_{in} = 2,5 + 0,5 \sin(\omega t)$  ( $\omega = 2\pi f$ ,  $f=1\text{kHz}$ ), y se alimenta con una fuente de 5 Voltios. Calcular  $R_B$  y  $R_C$  de forma que la señal de salida oscile en torno a  $2,5\text{V}$  al igual que la entrada. ¿Qué debemos hacer para que llegue al corte? ¿Y para que entre en saturación?



- E.2.7.** Dado el circuito de la figura, construir su característica de transferencia. ¿Invierte la señal?. ¿Por qué?. Supóngase  $\beta=250$ ,  $R_E=1k\Omega$  y  $R_B=10k\Omega$



- E.2.8.** Obtener por simulación las curvas características de un transistor MOS canal N trabajando en modo de realce de forma similar a la empleada para el transistor bipolar 2N2222A del ejercicio anterior. Obtener a partir de estas curvas el valor de la tensión umbral del dispositivo.



## Preparación de la Evaluación

*Hemos descompuesto el tema en tres grandes objetivos:*

### Objetivos

- 1:** *Unión P-N*
- 2:** *Transistores Bipolares*
- 3:** *Transistores MOS*

*Para cada uno de estos objetivos existen problemas resueltos en el texto de problemas que recomendamos intentar reproducir antes de empezar la preparación a la evaluación. Aquí vamos a introducir otros problemas análogos y algunas cuestiones teóricas que pueden ayudarnos a evaluar nuestro grado de comprensión de los aspectos básicos en diodos y transistores, tal como van a ser usados en los dos temas siguientes. En unos casos damos nosotros la solución. En otros casos dejamos la cuestión para que la resuelva el alumno. Entonces damos ciertas “pistas” sobre el método de solución.*

### 1. Unión P-N

- 1.1.** *¿Podrías describir cualitativamente el origen de la barrera de potencial que hace que una unión PN conduzca más en un sentido que en el opuesto?*

#### **Solución:**

El origen de la barrera está en la diferente composición del semiconductor extrínseco tipo P de la del tipo N. En el primero hay un exceso de huecos y en el segundo un exceso de electrones. Al “ponerlos en contacto”, aparece un proceso de difusión de portadores móviles que pasan desde donde hay más hasta donde hay menos y dejan un conjunto de cargas iónicas inmóviles (negativas en P y positivas en N) que dan origen a una barrera electrostática,  $V_d$ , cuya altura es tal que compensa el proceso de difusión.

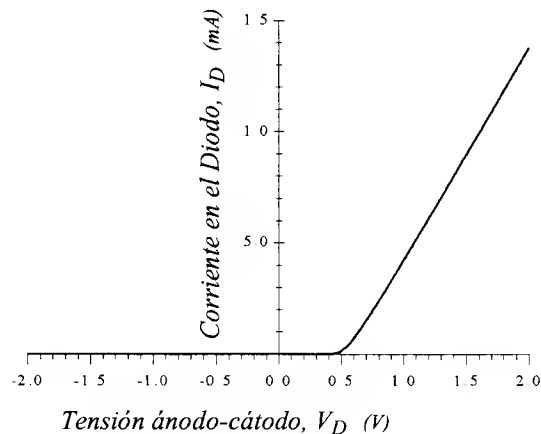
**1.2.** ¿Qué son las curvas características de un diodo?. ¿Cómo podríamos obtenerlas con el simulador?

**Solución:**

Son la representación gráfica de su comportamiento externo. Es decir, de la relación entre los valores de tensión en sus extremos,  $v_D = v_{\text{ánodo}} - v_{\text{cátodo}}$  y la corriente que pasa por el diodo,  $i_D$ . Analíticamente esta relación es de tipo exponencial:

$$i_D = I_S \left( e^{\frac{qv_D}{KT}} - 1 \right) \approx I_S e^{\frac{qv_D}{KT}}$$

pero puede aproximarse por dos tramos rectos: uno de poca pendiente (alta resistencia) y otro de gran pendiente (baja resistencia). El primero corresponde a la zona de corte, cuando el diodo no conduce y el segundo corresponde a la zona de saturación, cuando el diodo conduce con una resistencia muy baja, tal como ilustra la figura adjunta en la que se representa la curva característica del diodo 1N4148 obtenida por simulación.



Si el alumno tiene dificultades para obtener con el simulador esta curva característica le sugerimos que revise el circuito de la figura 2.1.1 del texto de problemas.

**1.3.** ¿Qué significa polarizar un diodo?. ¿Cómo podríamos usar el simulador para ilustrar los distintos valores de tensión y corriente que se obtienen al polarizar el diodo en distintos cuadrantes (1º y 3º)?

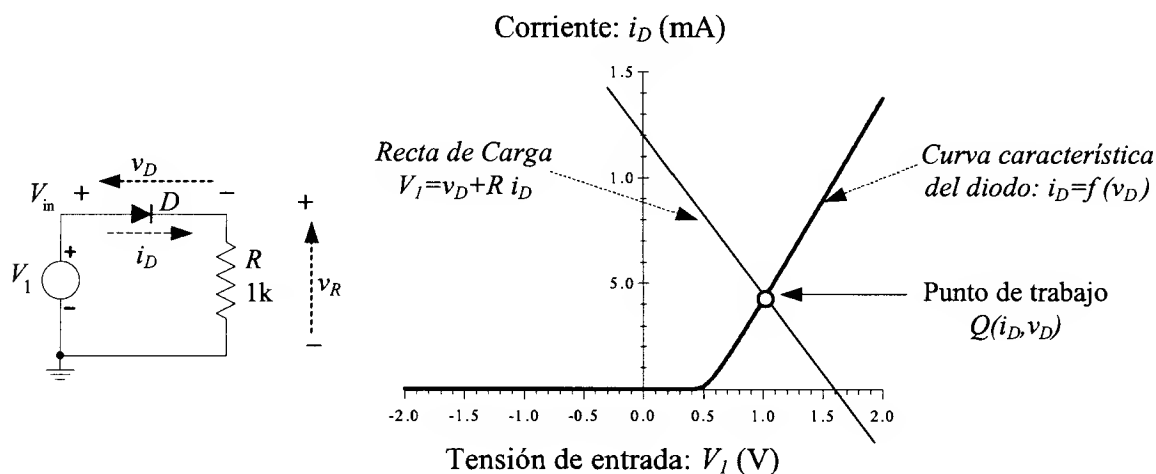
**Solución:**

Polarizar un dispositivo es elegir un punto de funcionamiento estático y hacer que se cumpla. Para ello, el diodo se introduce en un sencillo circuito con una pila y una resistencia, tal como se ilustra en la figura adjunta, de forma que los valores de tensión en los extremos del diodo deben satisfacer dos condiciones:

1. Las de la curva característica,  $i_D = f(v_D)$ , al mirar hacia el diodo
2. Las de la ley de Ohm, al mirar hacia la pila:  $V_I = v_D + R i_D$

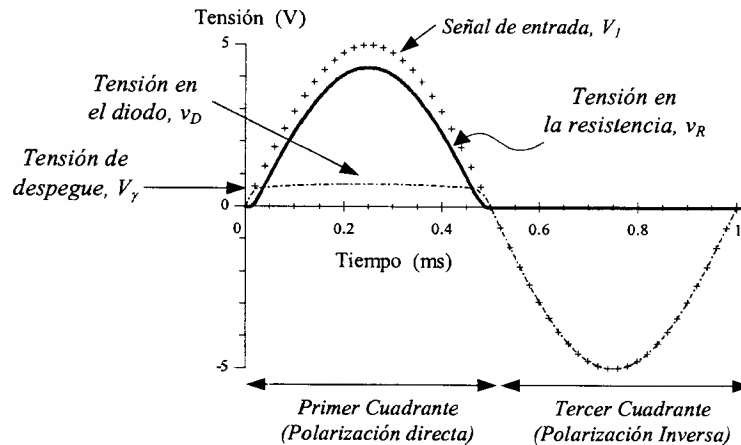
A esta condición que, como vemos corresponde a la ecuación de una recta se la llama Recta de Carga.

Así pues, para saber donde está situado funcionalmente el diodo, tenemos que resolver este sistema de dos ecuaciones con dos incógnitas y la forma de resolverlo más sencilla es de naturaleza gráfica. El punto de trabajo del diodo se encuentra en la intersección de la recta de carga con la curva característica.



Para ilustrar la polarización con el simulador es cómodo usar el circuito elemental anterior sustituyendo la pila (continua) por una señal alterna (sinusoidal) de baja frecuencia, que recorra los valores en los que el diodo conduce (primer cuadrante a partir de la tensión de despegue,  $V_\gamma$ , del orden de 0,6 voltios) y los valores en los que no conduce (semiciclos negativos).





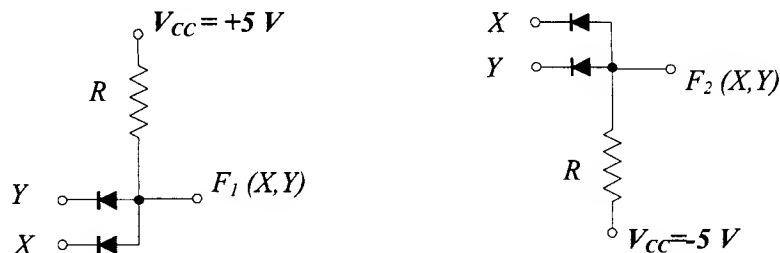
- 1.4. ¿Qué modelo analítico puede obtenerse de la simple inspección de las curvas características en el primer cuadrante?

(Recuerde la posibilidad de aproximar la curva por sus tangentes, dando lugar al modelo simplificado consistente en dos rectas que se cruzan en el valor de la tensión de despegue del diodo).

- 1.5. ¿Qué modelo analítico puede obtenerse de la siguiente descripción?: Un diodo polarizado en inversa no conduce (salvo una pequeñísima corriente de pérdidas) hasta que la tensión aplicada no alcanza un cierto valor de ruptura, llamado tensión Zener,  $V_Z$ . Entonces conduce con una resistencia muy baja,  $r_Z$ ).

(Recuerde aquí también la posibilidad de aproximar la curva característica del diodo zener en el cuarto cuadrante mediante dos rectas que se cruzan en el valor de la tensión zener,  $v_Z$ ).

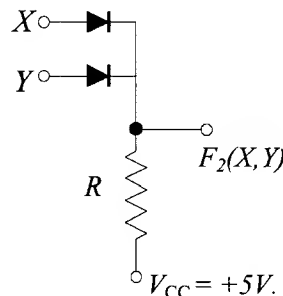
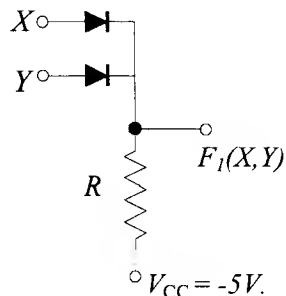
- 1.6. ¿Cómo se comportan los dos circuitos de la figura?. ¿Qué relación le encuentra con los operadores lógicos estudiados en el primer capítulo?. ¿Qué pasa al cambiar el signo de la pila en ambos circuitos?



El procedimiento para resolver la primera parte de esta cuestión es considerar que en las entradas  $X$  e  $Y$  se aplican tensiones de 0 voltios o de 5 voltios, correspondientes a los niveles lógicos “0” y “1”. Después, ante cada una de las cuatro configuraciones lógicas posibles, hay que ver cómo quedan polarizados los dos diodos (en directa o en inversa) y, consecuentemente, cuál es el valor de tensión en los cátodos,  $F_1(X,Y)$ .

Para el segundo circuito dejamos que el alumno razone teniendo en cuenta que se ha cambiado el valor de la tensión de alimentación y que recordamos la diferencia entre lógica positiva y lógica negativa.

**1.7.** *¿Sabría analizar los circuitos de la figura?. ¿Qué función lógica realizan?. ¿En qué tipo de lógica (positiva o negativa)?.*



## 2. Transistores Bipolares

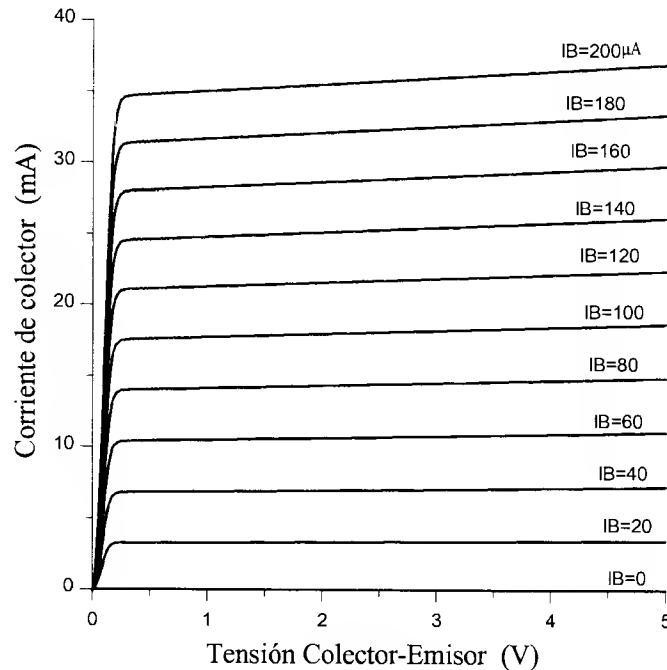
**2.1.** *¿Podría explicar cualitativamente el principio de funcionamiento del transistor bipolar haciendo uso de la figura 2.16 en la que se muestra el diagrama de corrientes?*

Le recordamos que si tiene dificultades puede ver el apartado 2.5.1 de este tema. Le recordamos también que el procedimiento de análisis cualitativo consiste en el seguimiento de la corriente desde el terminal de Emisor hasta los otros dos terminales (Base y Colector), llevando la contabilidad de los portadores que “mueren” por recombinación en las fronteras de las zonas de transición o durante el tránsito de la zona neutra de la región de Base. Finalmente, conviene también recordar que el análisis se hace en “modo activo normal”, es decir cuando la unión de Emisor está polarizada en sentido directo y la unión de Colector está polarizada en sentido inverso.

- 2.2. ¿Cómo podríamos obtener las características de entrada de un transistor bipolar en configuración de emisor común  $I_B=f(V_{BE})$ ? ¿A qué se parece el comportamiento de la unión Base-Emisor cuando el transistor está funcionando en modo activo normal?

Sugerimos el uso de la versión libre del PSpice y emplear el transistor 2N2222 y dos generadores de tensión tipo VSCR (fuentes de tensión continua) para poder comparar sus resultados con los que contiene el problema análogo del texto de problemas.

- 2.3. Si ya tenemos las características de entrada ¿cómo podríamos aprovechar el esfuerzo de simulación realizado en el ejercicio 2.2 para obtener ahora las características de salida  $I_C=f(V_{CE})$  del mismo transistor 2N2222? Acceder al problema E.2.5 del texto de problemas si se tiene dificultad en obtener las curvas análogas a las de la figura adjunta.



Características de salida del transistor 2N2222 obtenidas por simulación

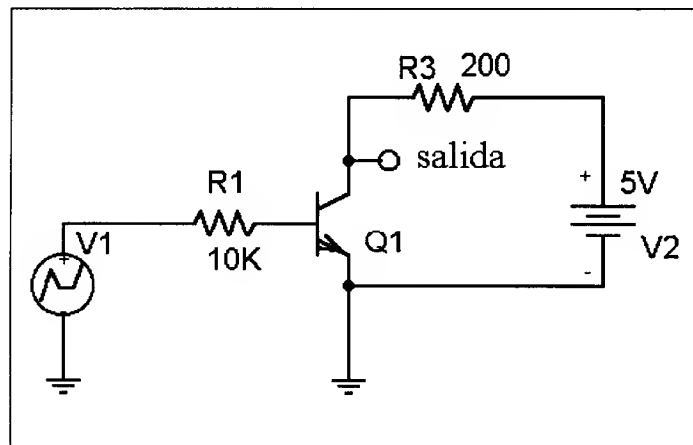
- 2.4. ¿Qué trayectoria sigue el punto de trabajo sobre la recta de carga al pasar de activa a corte? ¿Y al pasar de corte a saturación?

**2.5.** ¿Podría obtener y justificar la forma de onda en el Colector en transiciones del transistor de corte a saturación y viceversa? ¿Puede estimar con el simulador algún valor aproximado para los tiempos de retardo, subida, almacenamiento y caída? Por si le fuera de ayuda, el circuito y las formas de onda cualitativas correspondientes a las transiciones corte-saturación-corte las tiene en la figura 2.24 de este tema.

### **Solución:**

El objetivo de esta cuestión es ilustrar el comportamiento transitorio y los retardos de conmutación responsables finales de la limitación en la velocidad de conmutación de toda familia lógica.

Empezamos con el circuito inversor mínimo, consistente en un transistor polarizado con una resistencia de Colector ( $R_3 = 200 \, \Omega$ ) y una pila de 5 voltios ( $V_2$ ). A la entrada por Base aplicamos una onda cuadrada (o un pulso) de 5 voltios de amplitud, procedente en principio de la salida de otro inversor análogo y simulado con la fuente  $V_1$ .



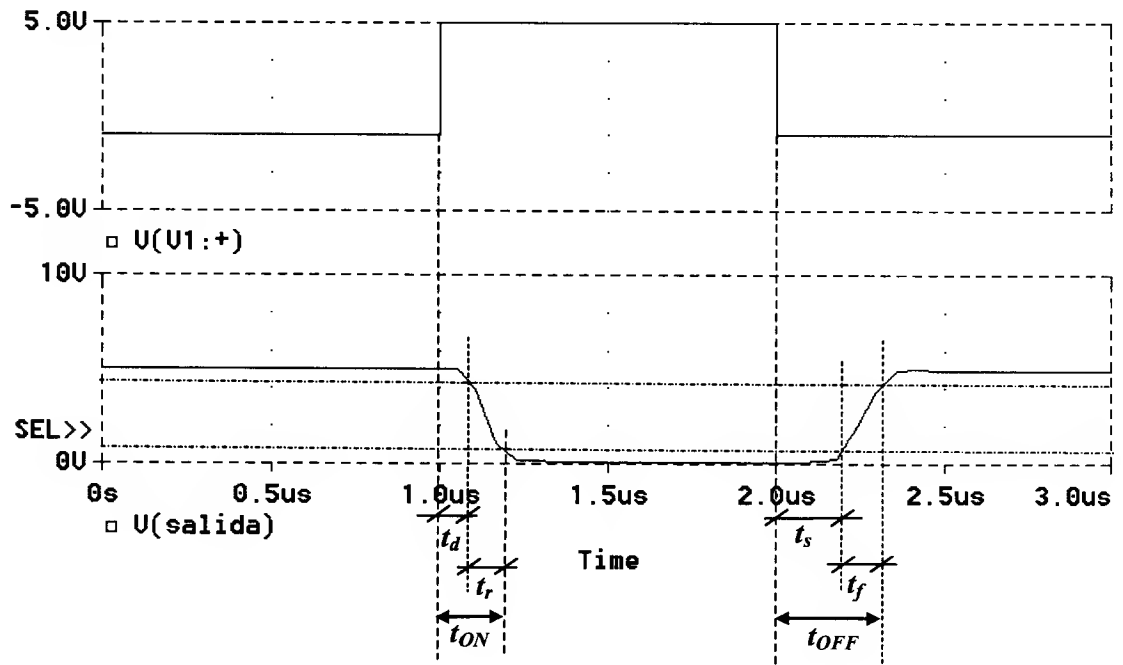
De la simple inspección de los resultados, se observan los dos procesos básicos responsables del valor finito de  $t_{ON}$  y  $t_{OFF}$ . Por una parte, cuando la entrada pasa de baja a alta, la salida no responde instantáneamente sino que tarda un cierto tiempo en iniciar al descenso (*tiempo de retardo*,  $t_d = 75ns$ ). Por otro lado, el descenso no es instantáneo sino que se realiza con cierta pendiente (derivada) finita que hace que tarde otro tiempo en alcanzar una banda estrecha en torno al “0” (*tiempo de subida*,  $t_r = 122ns$ ). La suma de estos dos tiempos da lugar al retardo en el paso a conducción.

$$t_{ON} = t_d + t_r = 75ns + 122ns = 197ns$$

Un proceso análogo ocurre en el paso al corte, dando lugar a un nuevo retardo

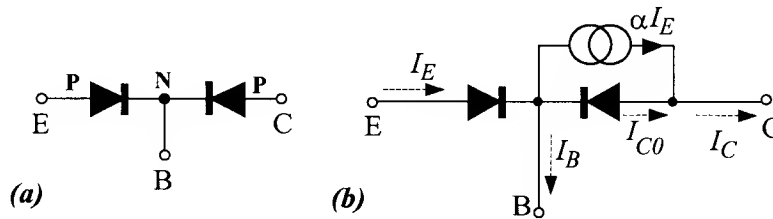
$$t_{OFF} = t_s + t_f = 146ns + 188ns = 334ns$$

Si tiene dificultad en comprender el origen y el significado físico de estos cuatro tiempos ( $t_d$ ,  $t_r$ ,  $t_s$  y  $t_f$ ) le recomendamos se lea el apartado 2.6 este tema.



$$t_d = 75ns \quad t_r = 122ns \quad t_{ON} = 197ns \quad t_s = 188ns \quad t_f = 146ns \quad t_{OFF} = 334ns$$

2.6. ¿Qué distingue el modelo de la parte (a) de la figura (dos diodos en oposición) del modelo de la parte (b) (los mismos dos pero con un generador de corriente en paralelo)? ¿Qué aspecto de la geometría de un transistor bipolar es el responsable de esta diferencia?

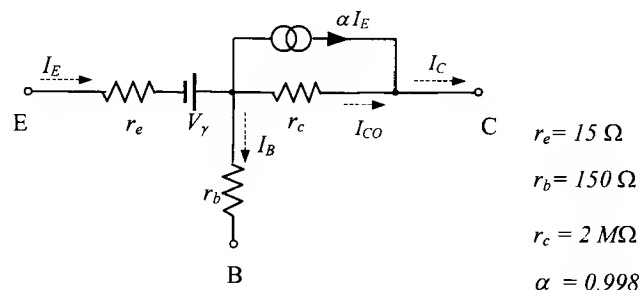
**Solución:**

En la parte **(a)** tenemos simplemente dos diodos en oposición en los que las corrientes que los recorren no son interdependientes. No hay por consiguiente efecto transistor; no hay amplificación.

En cambio, en la parte **(b)** vemos que en paralelo con el diodo de la unión de Colector hay un generador de corriente cuyo valor en todo momento,  $\alpha I_E$ , es función del valor de la corriente en el otro diodo,  $I_E$ . Así la corriente total en el colector,  $I_C = \alpha I_E + I_{CO}$ , es igual a la que genera el propio diodo de colector (polarizado en inversa y por consiguiente pequeña,  $I_{CO}$ ) más la que le viene inyectada desde la unión de emisor, que está polarizada en sentido directo. Por eso, este segundo modelo (b), corresponde a un transistor.

El aspecto geométrico que permite pasar de dos diodos en oposición a un transistor es el ancho de la zona de base. Cuando la base es muy estrecha, los portadores que la atraviesan desde la unión de emisor ( $I_E$ ) a la de colector no tienen tiempo de “morir” por recombinación y llegar casi intactos ( $\alpha I_E$ ). Si la zona de base fuera más ancha, se recombinarían y se perdería el efecto transistor, quedándonos reducidos al modelo de la parte (a) (dos diodos en oposición).

**2.7.** Recordando los modelos de un diodo polarizado en directa y en inversa ¿podría pasar del circuito de la parte (b) del ejercicio anterior al que mostramos ahora?. ¿Se le ha añadido algún componente que no proceda de los diodos?. ¿Qué significado físico tiene?.



(Recuerde los modelos del diodo como elemento de circuito en directa y en inversa)

- 2.8. *¿Qué razonamientos adicionales serían necesarios para pasar del modelo (b) del ejercicio 2.6. al más completo y simétrico que aparece en la figura 2.18 del tema?. ¿Qué nuevos componentes fenomenológicos le han sido añadidos y por qué?.*
- 2.9. *¿Cómo podríamos resumir con valores numéricos el modelo más simple del transistor bipolar en activa, corte y saturación?. ¿Por qué puede ser útil este resumen?.*

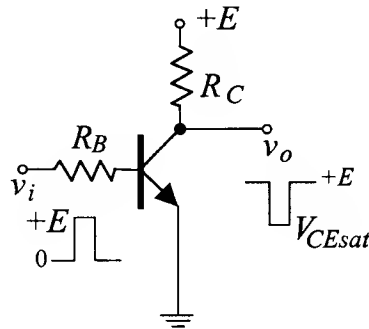
**Solución:**

MODO	Unión Base-Emisor		Unión Colector-Emisor	
	$V_{BE}$	$I_B$	$V_{CE}$	$I_C$
<b>Activo</b>	$0,6 \leq V_{BE} \leq 0,8$ voltios	<i>pequeña</i>	<i>grande</i> $V_{CE} = E - R_C I_C$	<i>grande</i> $I_C = \beta I_B$ $\beta \approx 300$
<b>Corte</b>	$V_{BE} \leq 0,6$ voltios	<i>despreciable</i>	$V_{CE} = E$	<i>despreciable</i>
<b>Saturación</b>	$V_{BE} \geq 0,8$ voltios	<i>pequeña</i>	$V_{CE} = 0,2$ voltios	<i>grande</i>

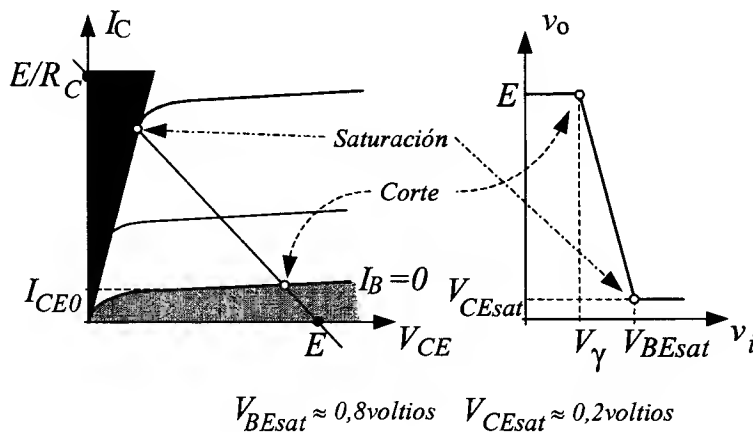
También podemos verlo sobre las curvas características, donde se marcan las zonas de trabajo. La línea elipsoidal de color pretende ilustrar la cadena causal de sucesos en la figura 2.20. Primero aplicamos una tensión,  $V_{BE}$ , en la unión de emisor que, de acuerdo con su valor ( $\leq$  ó  $>$  que  $0,6$  voltios), hace que la unión esté en corte, activa o saturación. Como consecuencia de esta situación en las características de entrada se genera una corriente,  $I_B$ , que hace que en la unión de colector tengamos  $I_C = \beta I_B$  y  $V_{CE} = E - R_C I_C$ . Gráficamente estamos diciendo que la  $I_B$  generada en la unión de emisor selecciona una curva característica en la salida y allí hay que ir a buscar las relaciones entre  $I_C$  y  $V_{CE}$ .

- 2.10. *Incidamos de nuevo en los conocimientos mínimos necesarios para comprender el funcionamiento en corte y saturación. ¿Podría explicar la situación que se ilustra en el circuito de la figura, donde un pulso positivo de altura  $E$  actúa sobre la unión Base-Emisor?*

**Solución:**



La trayectoria del punto de trabajo pasa del corte (porque  $v_i = 0$  voltios y por tanto  $V_i < V_\gamma$ ) a la saturación (porque  $v_i = E$  y por tanto es mayor que la tensión de saturación  $V_{BE\ sat} \approx 0,8$  voltios). Es decir, pasa desde  $I_B \approx 0$ ,  $I_C \approx I_{CE0}$  y  $V_{BE} \approx E$  hasta  $I_B = \text{valor alto}$ ,  $I_C = \beta I_B$  y  $V_{CE} = V_{CE\ sat} \approx 0,2$  voltios, tal como ilustra la curva de transferencia ( $v_i$   $v_o$ ) de la figura adjunta.



El valor de la corriente de saturación depende de la resistencia de carga  $R_C$  y del valor de la pila que hemos usado en polarización,  $E$ . Entonces, los dos estados estables en lógica saturada son:

$$\begin{array}{ll}
 \textbf{Corte} \left\{ \begin{array}{l} I_C = I_{CE0} \\ I_C = I_{Csat} \approx E/R_C \end{array} \right. & \textbf{Saturación} \left\{ \begin{array}{l} V_{CE} \approx E \\ V_{CE} = V_{CEsat} \approx 0,2V \end{array} \right.
 \end{array}$$



### 3. Transistores de Efecto Campo (MOS)

Por razones pedagógicas y para simplificar la ayuda a la evaluación en este objetivo, haremos referencia sólo a los transistores MOS que trabajan en modo de realce. No mencionaremos el modo de vaciamiento. Sin embargo, sí que distinguiremos entre realce canal P y realce canal N, porque estos dos tipos de transistores MOS complementarios se usan en la familia de puertas lógicas CMOS.

**3.1.** *¿Podría describir de forma cualitativa el funcionamiento de una estructura MIS en situación de acumulación, vaciamiento e inversión?*

(Recuerde que partimos de una situación en la que el potencial en la superficie es nulo y aplicamos una tensión externa de polarización que modifica la concentración de portadores en la superficie en relación con el valor inalterable de esa concentración al penetrar en el volumen del semiconductor).

**3.2.** *Podría justificar el concepto de transistor MOS como yuxtaposición de muchas estructuras MIS, como las cartas de una baraja?*

Puede apoyarse en la figura 2.33 que muestra la estructura física de un transistor MOS canal N.

**3.3.** *Cuál es la forma de las curvas características de un transistor MOS?. ¿Cuántas regiones se pueden distinguir?. ¿Tiene aquí el concepto “saturación” el mismo significado que en los transistores bipolares?.*

(Si tiene dificultades debe ir al apartado 2.9 de este tema y al problema E.2.8 del texto de problemas).

**3.4.** *¿Podría reproducir los resultados del problema E.2.8 y obtener las curvas características de un MOS de canal N?. Se le sugiere el transistor IRF150 de la biblioteca de la versión de evaluación del simulador PSpice.*

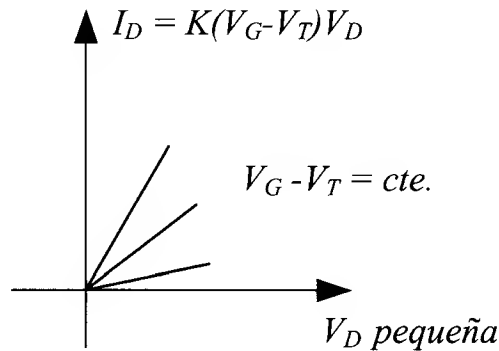
**3.5.** *La figura 2.35 muestra la forma cualitativa de las curvas características,  $I_D = f(V_D)$  para distintos valores de la función de puerta  $V_G$ , y en el desarrollo del tema propusimos una expresión analítica aproximada para estas curvas*

$$I_D = K \left[ (V_G - V_T) V_D - \frac{V_D^2}{2} \right]$$

Donde  $V_T$  es la tensión y  $K$  una constante función de la geometría del dispositivo. ¿Podría ahora avanzar un paso más en la simplificación y proponer una expresión para la corriente de drenador,  $I_D$  en la zona lineal y otra para la zona de saturación? ¿Qué pasa en la región intermedia?.

### Solución:

En zona lineal, despreciamos el término cuadrático y tenemos la ecuación de una recta  $I_D = K(V_G - V_T) V_D$  donde la pendiente,  $K(V_G - V_T)$ , depende de la diferencia entre el valor de la tensión de puerta ( $V_G$ ) y la tensión umbral ( $V_T$ ).



En zona de saturación, domina el término cuadrático y la corriente  $I_D$  permanece prácticamente constante e igual al último valor (el máximo) de la región anterior. Esto ocurre para un valor de  $V_D$  al que llamamos valor de saturación,  $V_{DS}$ , en el que si seguimos aumentando  $V_D$ ,  $I_D$  ya no aumenta. Es decir, su derivada es cero.

$$\left. \frac{\partial I_D}{\partial V_D} \right|_{V_G = cte} = 0 = K[(V_G - V_T) - V_{DS}]$$

luego  $V_{DSsat} = V_G - V_T$  y el valor de la corriente de saturación es:

$$I_{Dsat} = \frac{k}{2}(V_G - V_T)^2 = \frac{k}{2}V_{DSsat}^2$$

Veremos más adelante (tema 4) que con estas dos expresiones que ahora repetimos podemos comprender el comportamiento de las puertas MOS y CMOS.

<b><i>Zona Lineal</i></b>	$V_D$ pequeña	$I_D = K (V_G - V_T) V_D$
<b><i>Zona de Saturación</i></b>	$V_D > V_{DS}$ , con $V_{DSsat} = V_G - V_T$	$I_{Dsat} = \frac{K}{2} V_{DSsat}^2$

En la zona intermedia, entre la lineal y la de saturación, interviene la corrección cuadrática y hay que usar la expresión completa de  $I_D = f(V_D)$ .

**3.6.** *Ahora que conocemos las curvas características y las regiones de funcionamiento del MOS ¿podría proponer un experimento con el simulador para polarizar al transistor y para hacerlo actuar como un inversor?. ¿Podría repetir el ejercicio 2.6 para transistores MOS?. ¿Qué reflexiones le sugiere la comparación entre los valores de  $t_{ON}$  y  $t_{OFF}$  en ambos tipos de transistores?.*



Conviene recordar el carácter pedagógico de este tema orientado a la comprensión de los aspectos básicos del funcionamiento de los dispositivos como conmutadores. Para aquellos alumnos que quieran ampliar conocimientos se les sugiere ver la *bibliografía complementaria*, donde se aborda con mayor extensión y profundidad el estudio de los dispositivos electrónicos.

## REFERENCIAS BIBLIOGRÁFICAS

1. Carmona Suárez, E., Mira Mira, J.: “*Problemas Resueltos de Física de Dispositivos Electrónicos*”. UNED (ref. 07407). Madrid, 2000
2. Mira, J., Delgado, A.E.: “*Electrónica Analógica Lineal*” Tomo 1. UNED (ref. 074076). Madrid, 1993.
3. Sze, S.M.: “*Physics of Semiconductor Devices*”. J. Wiley & Sons. N.Y. 1981.



# TEMA 3

---

## ***Familias Lógicas (I): TTL***

---

### **+ CONTEXTO**

*El propósito general de este tema y el siguiente es estudiar las distintas familias lógicas. Aquí estudiamos la familia TTL (Lógica Transistor-Transistor) y en el tema 4 estudiaremos las familias ECL, MOS y CMOS. Después, volveremos a considerar las puertas lógicas en función de sus características externas y las usaremos para síntesis de funciones en lógica combinacional y secuencial.*

*En el primer tema estudiamos las exigencias computacionales del procesamiento digital de la información y llegamos al concepto de conjunto completo de operadores lógicos. Es decir, para la síntesis de cualquier función combinacional necesitamos circuitos AND, OR y NOT (inversores) o bien circuitos NAND o circuitos NOR.*

*En el segundo tema hemos presentado de forma resumida el comportamiento de diodos y transistores (bipolares y MOS) trabajando como conmutadores entre dos estados (corte y saturación). Tenemos así la base electrónica necesaria para el estudio de las distintas familias lógicas. Cada familia lógica corresponde a una forma específica de diseñar los operadores básicos.*

*Este tema nos proporciona los argumentos físicos para la caracterización de los circuitos lógicos en términos de parámetros tales como velocidad o consumo. Por consiguiente, es el puente entre “Electrónica Digital” y “Sistemas Digitales”, donde ya no se hace referencia a las estructuras internas de las puertas lógicas pero se depende de la solución tecnológica usada en la implementación. De hecho, una parte importante de las prestaciones de un sistema de cálculo dependen de la familia lógica (TTL, ECL ó CMOS, por ejemplo) usada en la síntesis de sus memorias y unidades de proceso.*

*En este capítulo estudiamos las bases de la integración en pequeña escala (SSI) que, en general, sólo posee utilidad pedagógica. Después estudiaremos funciones integradas en escala media (MSI) con más de cien puertas en un circuito integrado y en gran escala (LSI).*

## **++ CONOCIMIENTO PREVIO NECESARIO**

*Los conocimientos necesarios se encuentran en el tema 2. No vamos a usar otros conocimientos que los correspondientes a los niveles de corriente y tensión en los estados de corte y saturación de los transistores bipolares y la sencilla relación entre las corrientes de base y colector en zona lineal:  $I_C = \beta \cdot I_B$ . Después sólo necesitamos la Ley de Ohm,  $V_{CE} = E - I_C R_C$  y saber sumar las distintas caídas de tensión en las uniones y resistencias de una malla.*

## **+++ OBJETIVOS DEL TEMA**

**Objetivo 1:** *Saber caracterizar una familia lógica (características estáticas, transitorios, velocidad, ruido, consumo y flexibilidad lógica).*

**Objetivo 2:** *Comprender el funcionamiento de los circuitos RTL (Lógica Resistencia-Transistor) y DTL (Lógica Diodo-Transistor). Estos circuitos se introducen sólo por razones pedagógicas, porque son muy sencillos y constituyen la base de las familias CMOS y TTL.*

**Objetivo 3:** *Saber analizar en profundidad una puerta NAND en TTL y saber manejar datos de catálogo sobre puertas lógicas en TTL.*

## ++++ GUÍA DE ESTUDIO

*Este tema y el siguiente tienen cierta complejidad adicional para el lector con orientación a Sistemas Digitales, por lo que aconsejamos formarse primero una imagen clara sobre lo que es un transistor considerado como elemento de circuito en Electrónica Digital. Es decir, recordar que en esencia se trata de un conmutador que para ciertos valores de tensión entre base y emisor no conduce y pasado un cierto valor umbral conduce con muy baja resistencia. Es importante comprender el significado de la característica de transferencia y aconsejamos usar el simulador para seguir el camino de las señales desde las entradas de una puerta NAND, por ejemplo, hasta la salida. También aconsejamos volver a la teoría, a los problemas resueltos en el texto de problemas y a los ejercicios de preparación de la evaluación propuestos en el tema segundo cuando se encuentren dificultades.*



---

## ***Contenido del Tema***

---

### **3.1. INTRODUCCIÓN A LAS FAMILIAS LÓGICAS**

Cada familia se caracteriza por un conjunto de parámetros estáticos y dinámicos junto con otros factores tales como margen de ruido, consumo o valor de los niveles lógicos que las hacen más o menos adecuadas a la solución de un determinado problema de diseño, de acuerdo con las especificaciones funcionales de partida.

Las familias RTL (Lógica Resistencia-Transistor) y DTL (Lógica Diodo-Transistor) sólo se introducen con propósito pedagógico ya que ilustran la síntesis de una función lógica y los problemas de conexión entre puertas sin confusión de estados. Por otro lado, la familia DTL es el acceso conceptual a la TTL que sustituye sus diodos de entrada por un transistor multiemisor.

Cada familia se caracteriza por un conjunto de parámetros estáticos y dinámicos junto con otros factores tales como margen de ruido, consumo o valor de los niveles lógicos que las hacen más o menos adecuadas a la solución de un determinado problema de diseño, de acuerdo con las especificaciones funcionales de partida.

De entre las familias de validez actual hemos seleccionado la TTL (Lógica Transistor-Transistor ) como ejemplo de lógica saturada en tecnología bipolar, la ECL (Lógica de Emisores Acoplados), como familia rápida de lógica no saturada en tecnología bipolar y la CMOS (MOS-complementario) como familia representativa de la tecnología MOS. Existen otras familias, como la HTL (Lógica de Alto Umbral), para aplicaciones industriales que deben trabajar en entornos ruidosos) y la I<sup>2</sup>L (Lógica de Inyección) que no las vamos a considerar por limitaciones en la extensión del programa y en los objetivos pedagógicos aceptados para este material docente.

### **3.2. CARACTERIZACIÓN DE LAS DISTINTAS FAMILIAS LÓGICAS**

Aunque por razones pedagógicas estudiaremos el análisis y diseño de las distintas familias

mediante circuitos discretos, toda la electrónica digital está en tecnología integrada de forma que para nosotros un circuito digital integrado sólo es accesible a través de sus terminales de entrada y salida y para caracterizar su comportamiento eléctrico (estático y dinámico) necesitamos realizar medidas. Al no tener nosotros el control sobre el diseño interno, salvo en lógica programable, la única opción que nos queda es caracterizar su comportamiento externo. Las características básicas son las siguientes.

- ✧ *Características estáticas* (de transferencia, de entrada y de salida).
- ✧ *Transitorios* (tiempos de retardo, de subida y de bajada). Debe especificarse bajo qué condiciones de excitación y carga se realizan las medidas.
- ✧ *Ruido* (origen y margen de ruido que acepta un circuito sin que se produzca confusión de niveles lógicos)
- ✧ *Consumo* (función de la tecnología usada y dependiente en algunas familias de la velocidad deseada).
- ✧ *Flexibilidad lógica* (facilidad de interconexión con otras familias tras la necesaria adaptación de impedancias y niveles lógicos).

### 3.2.1 Características Estáticas

#### a) Características de Transferencia

Representan la relación entre la tensión de entrada y la de salida. En cada caso hay que especificar el número de entradas conectadas a “alta” (“1”) y a “baja” (“0”) y el número de circuitos que cargan a la puerta. La forma normal de calcular la curva de transferencia es suponer que la puerta lógica sobre la que estamos midiendo está conectada a la entrada y a la salida con otras puertas de la misma familia. La *figura 3.1.a* ilustra el esquema de medida y la *figura 3.1.b* el resultado obtenido con carácter general.

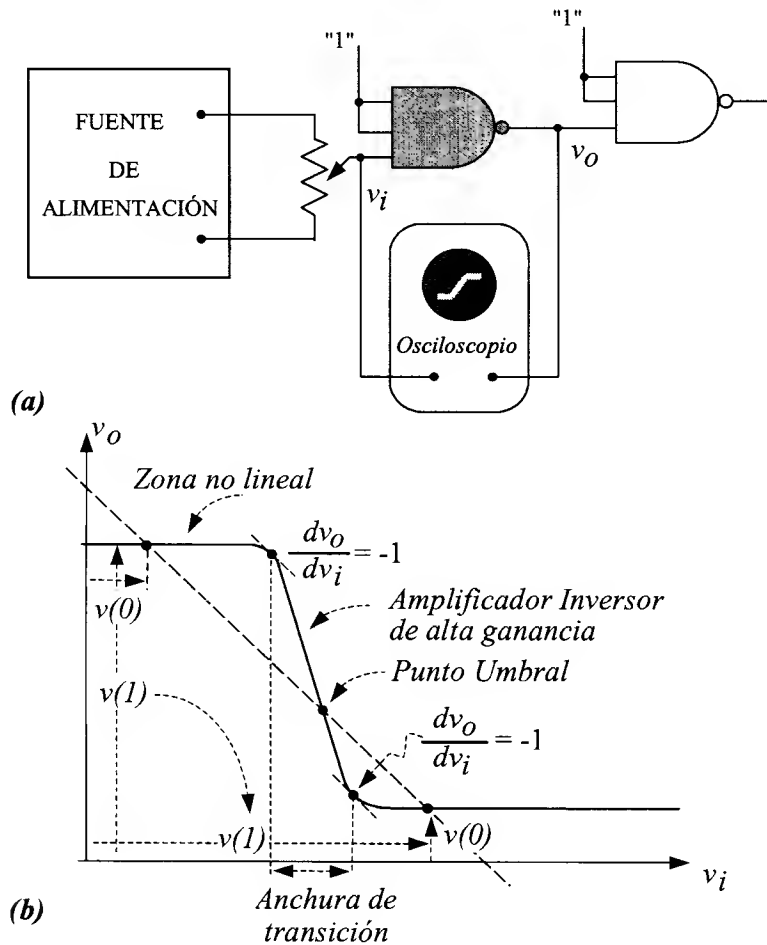
Veamos como se obtienen a partir de esta curva de transferencia los dos puntos de operación del inversor:

$$v_i = "0" \Rightarrow v_o = "1" \quad \text{y} \quad v_i = "1" \Rightarrow v_o = "0"$$

Esto equivale a calcular  $v(0)$  y  $v(1)$  que deben ser los mismos para la entrada que para la salida, ya que la salida de un circuito lógico debe ser compatible con su entrada. De hecho una tensión de



salida actúa como entrada en la etapa siguiente.



**Figura 3.1.** Característica de transferencia. (a) Circuito de medida. (b) Forma típica de la característica de transferencia.

Supongamos que la entrada lógica es "1" y veamos el punto de operación correspondiente. Para ello el nivel alto de la señal de salida se lleva sobre el eje de  $v_i$  determinando sobre la curva el punto de trabajo. Tenemos así el valor de "0" que llevado sobre el eje de  $v_o$  nos da sobre la curva de transferencia el valor de  $v_i$  que lo produciría y obtenemos así los dos puntos de trabajo. Sobre estas curvas de transferencia aparecen los siguientes parámetros de interés:

- *Niveles lógicos:*  $v(0)$  y  $v(1)$
- *Salto lógico:*  $v(1) - v(0)$
- *Punto umbral:* La recta que une ambos puntos de operación presenta una pendiente de -1 y corta a la característica de transferencia en un tercer punto llamado punto umbral.
- *Puntos de ganancia unidad:* son los dos puntos de la curva de transferencia en que la pendiente es -1 y definen la región de transición.
- *Anchura de transición:* es el cambio en la tensión de entrada necesario para variar la de salida desde el valor correspondiente al primer punto de ganancia unidad al valor correspondiente al segundo punto.

### b) Características de Entrada y Salida

Representan las tensiones de entrada (o salida) en función de las corrientes suministradas o absorbidas por la entrada (o salida). Se miden conectando puertas similares a la de prueba para tener en cuenta los efectos de la carga.

Lógicamente, hay una gran diversidad de curvas de entrada y salida. Sin embargo se suelen medir sobre una entrada dejando todas las demás a un nivel constante y se representan varias de ellas como función de la temperatura o del valor de la fuente de alimentación.

### 3.2.2. Transitorios.

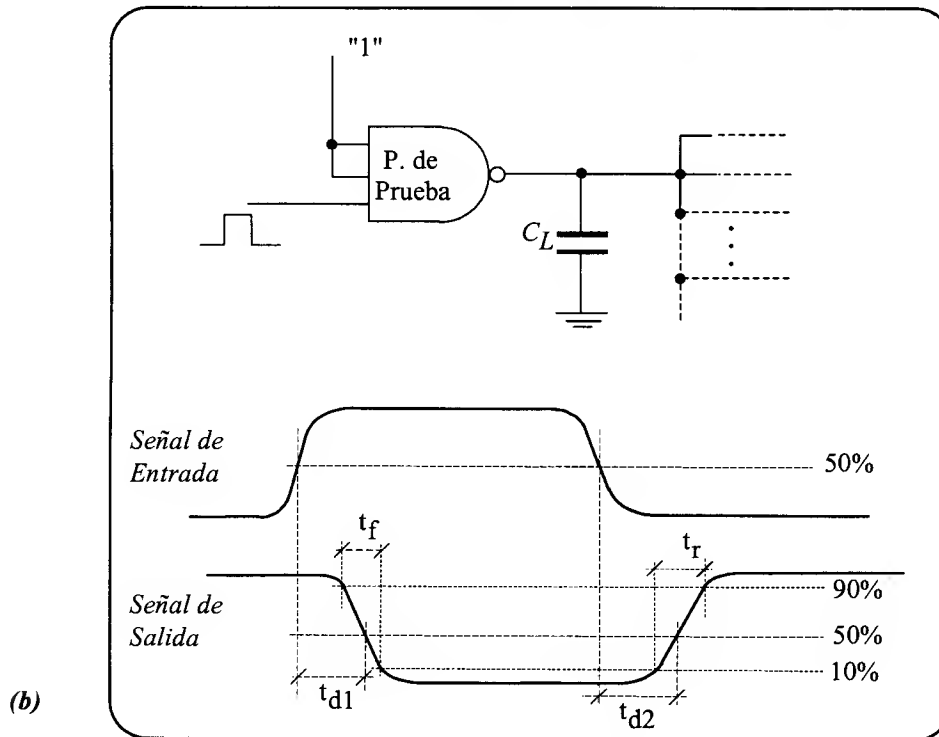
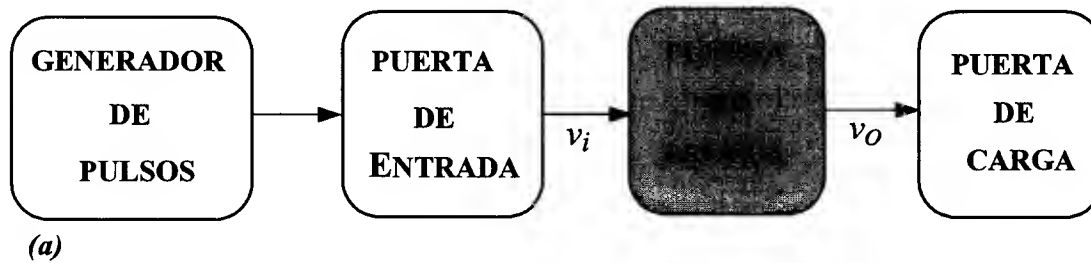
Miden los retardos en el paso de un estado a otro mediante los tiempos de retardo, subida y bajada. Para medirlos se usa una cadena de propagación, como la que se ilustra en la *figura 3.2.a*. Siempre la puerta de prueba posee las entradas no usadas conectadas al valor de alta para NAND.

El *tiempo de retardo* ( $\overline{t_d}$ ) se define como la media de los retardos correspondientes a la subida y a la bajada del impulso que constituye la tensión de entrada,

$$\overline{t_d} = \frac{t_{d1} + t_{d2}}{2}$$

[3.1]

siendo  $t_{d1}$  (para el caso de un circuito que invierte) el tiempo desde que la tensión de entrada alcanza el 50% de su valor final hasta que lo alcanza la salida. El tiempo  $t_{d2}$  se define de forma recíproca.



**Figura 3.2.** Medida de transitorios. (a) Circuito de medida. (b) Tiempos de retardo, subida y bajada..

Análogamente los *tiempos de subida* ( $t_r$ ) y *bajada* ( $t_f$ ) están asociados a los cambios de tensión entre el 10 y el 90 % ( $t_r$ ) y entre el 90 y el 10 % ( $t_f$ ). Una forma muy usual de comparar familias lógicas es usar el producto (potencia consumida x tiempo de retardo). Lógicamente, interesa que sea mínimo este producto puesto que entonces tendríamos la familia más rápida con el mínimo consumo. Se alcanzan excelentes resultados en el caso del CMOS dinámico.

### 3.2.3. Ruido

Ruido es toda señal no deseada que puede producir cambios en la salida o en el estado de un circuito digital que no corresponden al propósito de su diseñador. Hay distintos tipos de ruido de los que nos defendemos mediante los procedimientos típicos de: (1) apantallamientos (uso de cable coaxial y jaula), (2) vías a tierra y (3) desacoplos (filtros L-C).

Estos tipos de ruido pueden ser de origen externo o interno. Son de origen externo los debidos a la alimentación ( $V_{CC}$ ), disparos de triacs y otros conmutadores, etc... y se introducen casi siempre por acoplos capacitivos. Son de origen interno los generados por las impedancias parásitas y las espigas de corriente causadas por la conmutación de los propios circuitos lógicos.

El comportamiento de cualquier familia lógica frente al ruido se caracteriza con los siguientes parámetros: sensibilidad al ruido, inmunidad y rechazo del ruido. La inmunidad al ruido depende de muchos factores tales como: fuente de alimentación (caso de CMOS), las impedancias de entrada y salida, capacidades parásitas y dispersión en las curvas características, consecuencia del proceso de fabricación.

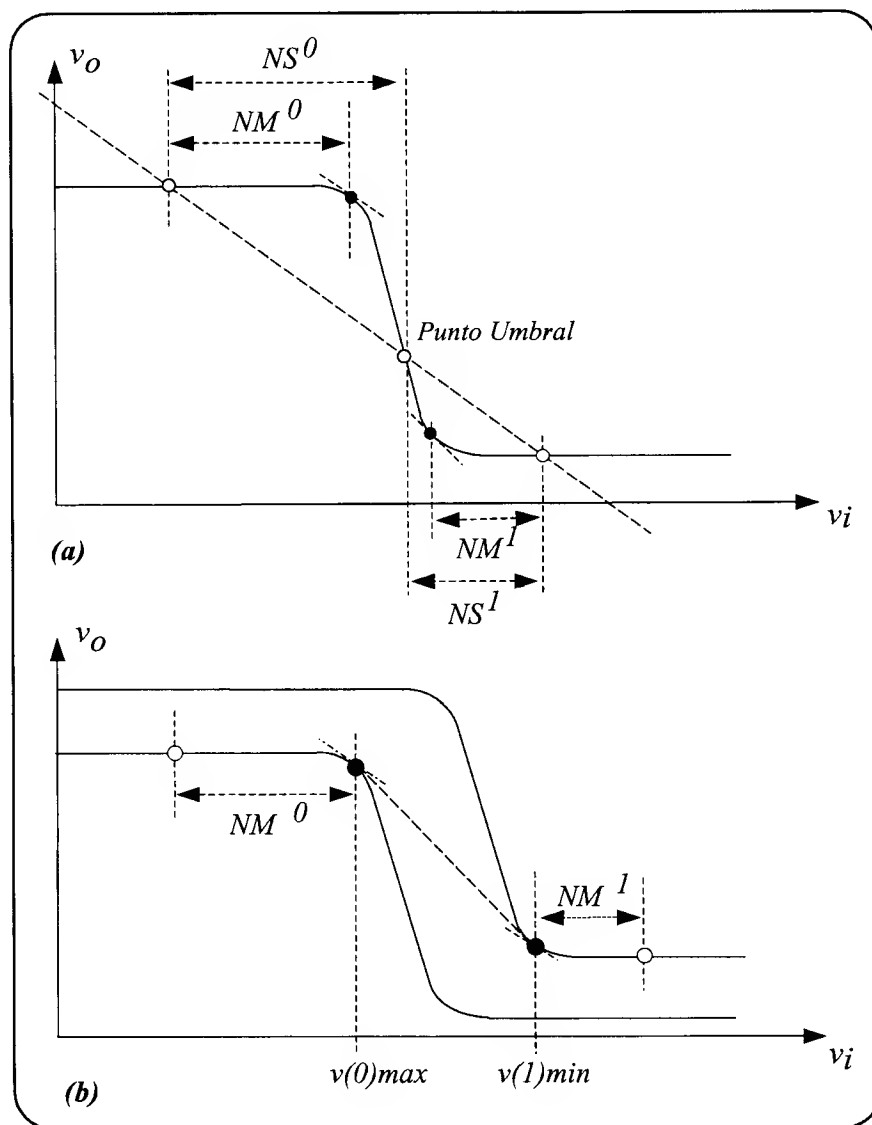
Los fabricantes suelen caracterizar la inmunidad al ruido de sus circuitos mediante:

1. *Tensión de ruido máxima tolerable.*
2. *Altura del impulso máximo tolerable.*
3. *Energía máxima que tolera el sistema sin producir un cambio indeseable en el nivel de salida.*

De todos estos parámetros el más general es la energía, pues permite una comparación entre las distintas familias. Como consideración general, antes de estudiar la inmunidad de una familia lógica, diremos que depende de su velocidad, puesto que las señales de ruido se unen con el mensaje vía acoplo capacitivo. Así, cuanto más lenta es una familia mayor es su inmunidad.

Las especificaciones de inmunidad al ruido en continua se pueden estudiar a partir de las características de transferencia. El margen de ruido ( $NM$ , noise margin) es la diferencia de tensiones, medida en el eje de  $v_i$ , entre el punto de funcionamiento y el de ganancia unidad más próximo.

Como en general es diferente para las entradas "1" y "0" distinguimos dos márgenes de ruido, el del nivel lógico "0" ( $NM^0$ ) y el del nivel lógico "1" ( $NM^1$ ) tal como se ilustra en la figura 3.3.a.



**Figura 3.3.** Ruido. (a) Margen de ruido en "0" y "1". (b) Zona de transición debida a la dispersión y puntos de cálculo "en el peor de los casos".

En el caso de la figura  $NM^0 > NM^1$  pero hay que indicar que las impedancias en uno y otro estado son distintas [ $Z(0) \neq Z(1)$ ] por lo que habrá que tenerlas en cuenta al analizar la susceptibilidad del circuito a la tensión de ruido.

Si las características de transferencia son de pendiente muy acusada, no vale la pena distinguir entre los dos puntos de ganancia unidad y podemos definir los márgenes de ruido en cada estado en función de la distancia a la tensión umbral. En este caso el parámetro se llama sensibilidad (*NS*, noise sensibility). Finalmente, se suele hablar de inmunidad al ruido (*NI*) como el cociente entre la sensibilidad y la amplitud lógica  $\{NS/[v(1)-v(0)]\}$ . Normalmente se usa sólo el parámetro *NM* sobre curvas de dispersión. Al considerar las características, hemos tomado una única curva de transferencia. La dispersión en características y los efectos debidos a las variaciones de temperatura, impedancias, derivas en  $V_{CC}$ , etc. hacen que sólo podamos hablar de una zona de transferencia lo que exige un método de diseño en el que nos tenemos que poner en el “peor de los casos” (*figura 3.3.b*). Esto exige, dado que no hay conmutación si la ganancia del inversor es menor que la unidad, que tomemos el primer punto en la frontera primera de la zona de transición y el segundo en la más externa. Vemos que existe un intervalo  $[v(1)_{min}-v(0)_{max}]$  en que la salida no está definida.

En cuanto a la inmunidad a pulsos podemos afirmar que a medida que el ancho del pulso se aproxima al tiempo de propagación, la amplitud del pulso necesaria para inducir a error aumenta mucho. Por otro lado, al hacer ancho el pulso el valor de la inmunidad se aproxima al margen para continua.

Finalmente, la inmunidad frente a la energía de la señal de ruido es considerada por muchos como la mejor especificación pues considera los efectos de las tensiones y las impedancias de forma conjunta.

### 3.2.4. Flexibilidad Lógica

Es una medida de la capacidad, versatilidad o variedad de uso de una familia lógica. Esto se valora viendo si tiene una ó más de las siguientes posibilidades:

- a) *Cableado lógico.*
- b) *Salidas complementarias.*
- c) *Capacidad de excitación (fan-out y open collector).*
- d) *Puertas múltiples (tipos de funciones).*
- e) *Posibilidad de usos alternativos o complementarios en Electrónica Analógica.*

El *cableado lógico* es la capacidad de realizar una función lógica a partir de la conexión externa

de puertas anteriores sin necesidad de emplear una puerta adicional.

La mayoría de las veces es necesario disponer de *una señal y su complementaria*. Si las puertas de la familia con la que estamos diseñando las ofrecen nos ahorramos un inversor.

Si la familia tiene capacidad para atacar a otras familias lógicas (*compatibilidad*) o cargas no estándar: visualizadores, relés, líneas de transmisión, etc. ello supondrá una reducción en costo, al no ser necesarios circuitos de acoplo. El problema de la capacidad de excitación puede estar condicionado a los valores de corriente o a la máxima excursión de la tensión de salida (series “open collector”, colector abierto). En general hacen falta circuitos de acoplo (hexinverters, drivers) y otros de tipo mixto (entrada digital y salida de potencia).

El “*fan-out*” es el número de puertas iguales que se pueden conectar a la salida de una dada sin perturbar su estado. Las salidas con “*colector abierto*” nacen de la necesidad de usar una resistencia de carga distinta de la de salida de la familia o una tensión de alimentación ( $V_{CC}$ ) mayor para encender lámparas, por ejemplo.

Todas las familias lógicas están dirigidas a usos digitales. Sin embargo, es posible usar las puertas en algunas *aplicaciones analógicas*. Además, en un circuito complejo con parte analógica y parte digital, si no queremos usar varias fuentes de alimentación ni tener problemas de adaptación de niveles podemos sintetizar las pocas funciones analógicas adicionales que incluye un sistema digital mediante los mismos circuitos digitales. Obsérvese que en la zona de transición, todo inversor es un amplificador de muy alta ganancia. Un ejemplo frecuente de uso de los circuitos digitales en el diseño de algunas funciones analógicas es el de los osciladores de cristal de cuarzo que incluyen los relojes digitales.

Finalmente, la *compatibilidad* de una familia nos dice con qué otras familias se puede unir. Los problemas fundamentales son los *distintos niveles lógicos* y las *distintas impedancias*. Sin embargo siempre existe una posibilidad de diseñar un circuito de acoplo entre dos familias lógicas diferentes. Para ciertas familias de uso frecuente (TTL y CMOS), es usual que el propio circuito incluya salidas dobles, cada una de ellas compatible con los niveles e impedancias de una familia concreta.

La tabla adjunta resume los distintos parámetros que caracterizan a una familia lógica con una descripción resumida de los distintos conceptos que sirven para evaluar cada una de estas características.

Caracterización de las distintas familias lógicas		
✦ Características Estáticas	✧ Características de Transferencia	✧ Niveles lógicos ✧ Salto lógico ✧ Punto umbral ✧ Puntos de ganancia unidad ✧ Anchura de transición
	✧ Características de Entrada y Salida	
✦ Transitorios	✧ Tiempos de retardo ✧ Tiempo de subida ✧ Tiempo de bajada	
✦ Ruido	✧ Sensibilidad al ruido ✧ Rechazo del ruido ✧ Inmunidad    ✧ Margen de ruido en "0" y "1"	
✦ Consumo		
✦ Flexibilidad Lógica	✧ Compatibilidad y acoplo con otras familias ✧ Cableado lógico ✧ Salidas complementarias ✧ Capacidad de excitación (fan-out, open colector) ✧ Puertas múltiples ✧ Posibilidad de usos alternativos	

### 3.3 LÓGICA RESISTENCIA-TRANSISTOR

Ya vimos en el tema anterior el funcionamiento del inversor usando transistores bipolares. Recordemos que se trataba simplemente de una etapa amplificadora en emisor común con una elección adecuada de las resistencias  $R_B$  y  $R_C$ , para que cuando la señal en la base del transistor conmutaba entre los niveles lógicos  $v("0")$  y  $v("1")$ , hacía que la señal en el colector conmutara entre  $v("1")$  y  $v("0")$ , respectivamente. Así pues, el transistor trabajaba pasando del estado de corte [ $v_O = +V_{CC} = v("1")$ ] al de saturación [ $v_O \approx 0,2V = v("0")$ ] y viceversa.



La figura 3.4 ilustra el circuito correspondiente a la puerta “NOR” de tres entradas en RTL. Cada entrada usa un transistor y los colectores de los transistores están unidos y apoyados a la fuente de alimentación a través de  $R_C$ . Trabaja en lógica positiva, es decir  $v(“1”) = 1,8\text{ V}$  y  $v(“0”) = 0,2\text{ V}$ .

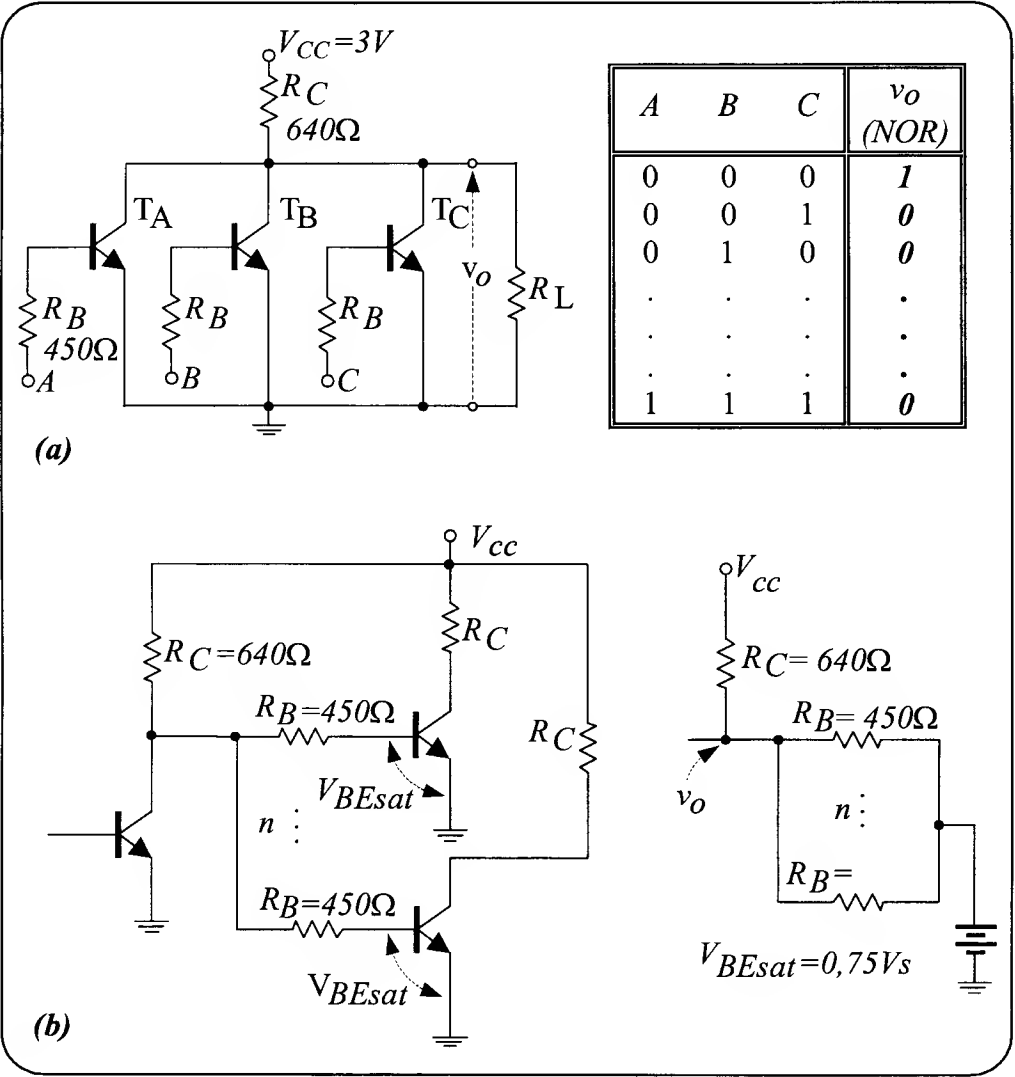


Figura. 3.4. Puerta NOR en RTL. (a) Circuito básico y tabla de verdad. (b) Ilustración del cálculo del “fan-out” y del margen de ruido.

Como se trata de una puerta NOR tendremos que comprobar su comportamiento en dos situaciones. Primero cuando las tres entradas son cero y, por consiguiente, la salida tiene que ser uno. Para el resto de las configuraciones en las que una o más de las entradas estén en alta, la salida debe ser cero.

Veamos primero que pasa cuando todas las entradas están en baja ( $A=B=C="0"$ ). Entonces los tres transistores,  $T_A$ ,  $T_B$  y  $T_C$ , están al corte y no hay contribución a la corriente que atraviesa  $R_C$  de ninguno de los tres colectores. Por consiguiente la tensión de salida,  $v_o$ , tiende a aproximarse a la tensión de alimentación ( $V_{CC}=3\text{ V}$ ).

$$v_o = V_{CC} - R_C \cdot I_C = v("1") \quad [3.2.]$$

Para conocer con precisión el valor final de la tensión de salida en el estado de alta necesitamos conocer cual es el número de puertas que se conectan. Cuantas más se conecten, menor será el valor de la impedancia paralelo correspondiente y, por consiguiente, mayor el valor de la corriente  $I_C$ . Esta variación en el valor de  $I_C$  hará que sea también mayor la caída en la resistencia de colector  $R_C$ . El límite se establece al evitar que se confunda el estado de alta.

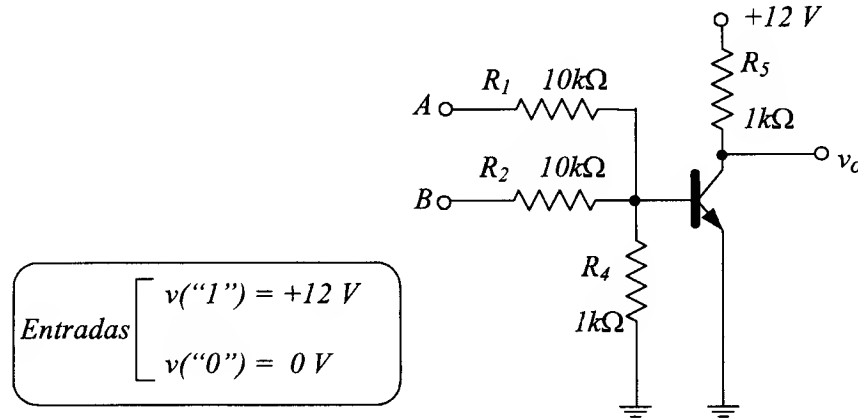
Por el contrario, para el resto de las configuraciones de entrada (desde 001 hasta 111), siempre hay una entrada en alta. Como bastan  $0,6\text{ V}$  para hacer conducir al transistor y  $v("1") = 1,8 > 0,6$ , el transistor correspondiente pasa a saturación y fija la tensión de salida a  $V_{CE(sat)} = 0,2\text{ V}$ , es decir a  $v("0")$  dando lugar al comportamiento NOR.

Este circuito presenta bajo valor de conectividad ("*fan-out*") y bajo margen de ruido, pero ilustra claramente el significado de estos parámetros (*figura 3.4.b*). Cada nueva puerta NOR en RTL que se conecte a la salida de una puerta de prueba contribuye con una vía a tierra formada por  $R_B$  y la unión base-emisor de un transistor que en saturación vale  $0,75\text{ V}$ . Por consiguiente, el efecto de la carga será,

$$v_o = V_{CC} - R_C \cdot I_C = V_{CC} - R_C \cdot \frac{V_{CC} - 0,75}{R_C + \frac{R_B}{n}} \quad [3.3]$$

Al aumentar  $n$ , disminuye  $v_o$  por lo que hay un valor máximo de  $n$  a partir del cual se confunde el estado. Esto hace que el margen de ruido de este circuito y su grado de conectividad sean muy bajos.

**Ejercicio:** Analizar el circuito de la figura y comprobar que es una puerta NOR si interpretamos como “1” lógico +12 V y como “0” lógico los 0 V.



**Figura 3.5.** Puerta NOR RTL y en lógica positiva.

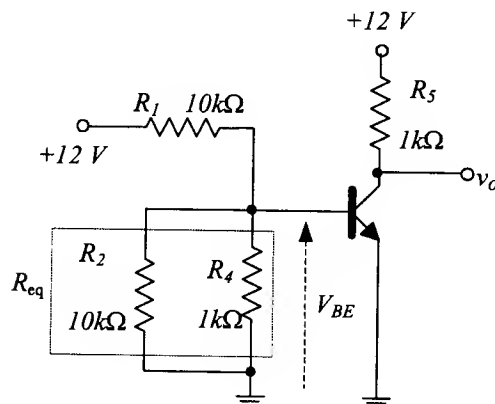
**Solución:**

Este circuito es una *puerta NOR para lógica positiva* porque basta con que una de las entradas esté a “1”, es decir que su tensión de entrada sea 12 V, para que la tensión Base-Emisor del transistor supere la tensión de saturación de dicha unión ( $V_{BE} \geq 0,8 \text{ V}$ ). En estas condiciones el transistor está en saturación y, por consiguiente, su tensión de Colector-Emisor es despreciable (siempre menor que 0,2 V), lo que corresponde al “0” lógico.

Solo en el caso en que las dos entradas estén a “0”, es decir que su tensión sea de 0 V, el transistor no conducirá, ya que la tensión Base-Emisor no supera la tensión de despegue de la unión. En esta caso la salida estará a 12 V, es decir en “1” lógico.

Podemos calcular analíticamente estas dos condiciones y, por consiguiente, obtener la tabla de verdad.

1. Ante 01 y 10 la configuración de entrada es la de la *figura 3.6*. Si despreciamos el pequeño valor de la corriente de Base podemos estimar el valor de la tensión Base-Emisor a través de un simple partidor de tensión ( $R_1, R_{eq}$ ) tal como se muestra en la figura. El valor obtenido (0,999 V) garantiza la saturación del transistor.

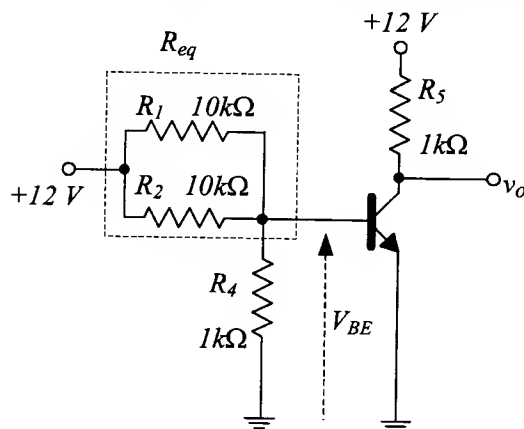


$$R_{eq} = \frac{R_2 \cdot R_4}{R_2 + R_4} = \frac{1k\Omega \cdot 10k\Omega}{1k\Omega + 10k\Omega} = 909\Omega$$

$$V_{BE} = 12 \frac{R_{eq}}{R_1 + R_{eq}} = 0,999V$$

**Figura 3.6.** Circuito correspondiente a las entradas 01 y 10.

2. Ante la configuración de entrada 11, el circuito es el que se muestra en la figura 3.7, con lo que de nuevo supera los 0,8 V y el transistor está en saturación.



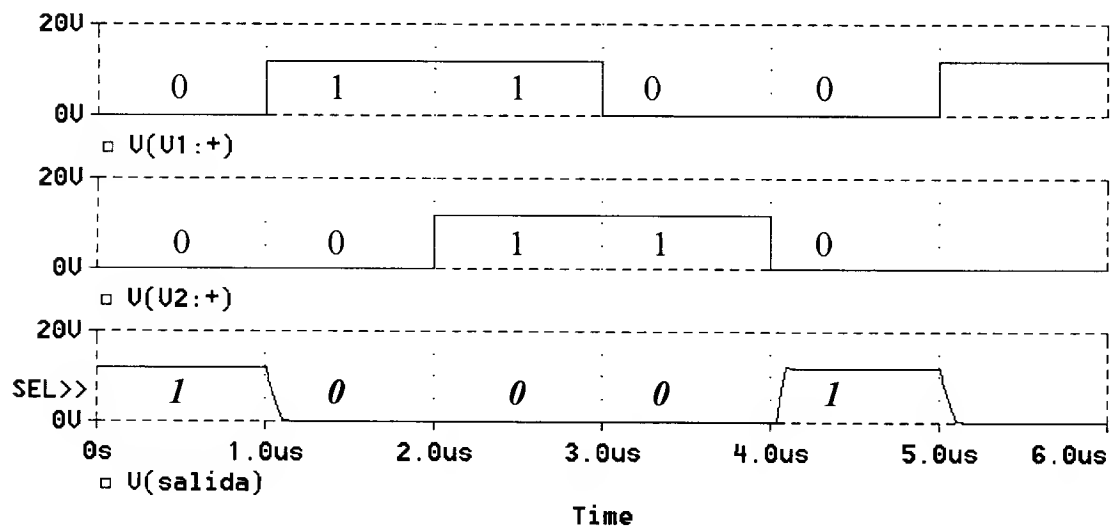
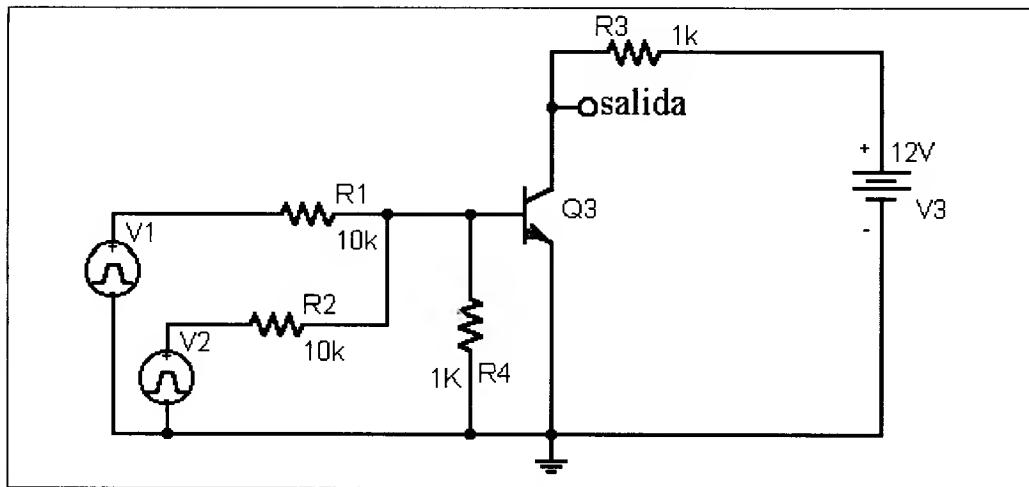
$$R_{eq} = \frac{R_1 \cdot R_2}{R_1 + R_2} = \frac{10k\Omega \cdot 10k\Omega}{10k\Omega + 10k\Omega} = 5k\Omega$$

$$V_{BE} = 12 \frac{R_4}{R_4 + R_{eq}} = 2V$$

**Figura 3.7.** Circuito correspondiente a la entrada 11

3. Ante la configuración de entrada 00, el circuito de entrada está formado por las tres resistencias en paralelo ( $R_1$ ,  $R_2$  y  $R_4$ ) uniendo la base a tierra y, obviamente,  $V_{BE} = 0$ , el transistor no conduce ( $I_C = 0$ ) y la tensión de salida es:  $v_o = 12 - I_C R_C = 12V$ .

Podemos ahora usar el simulador para comprobar que el circuito se comporta efectivamente como una puerta NOR (figura 3.8).



**Figura 3.8.** Simulación y obtención de la tabla de verdad de una puerta NOR.

**Ejercicio:** *¿Qué modificaciones tendríamos que introducir en el circuito NOR del ejercicio anterior para convertirlo en una puerta NAND?*

**Solución:**

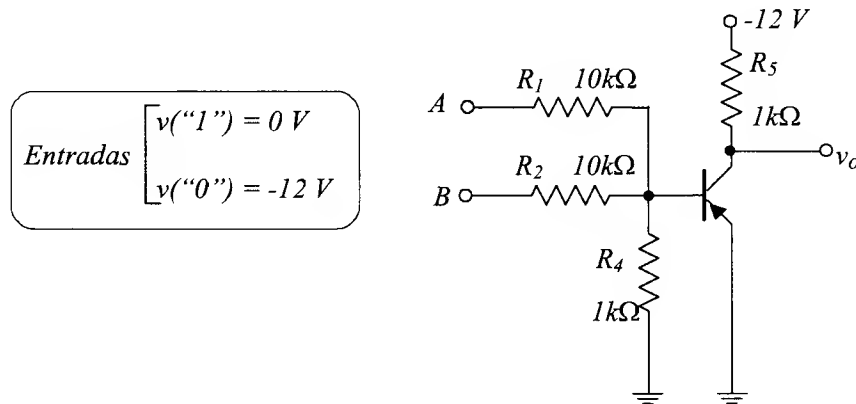
Para convertir este circuito de NOR a NAND tendríamos que realizar las siguientes modificaciones:

1. Pasar de transistor NPN a PNP y polarizarlo con  $-12\text{ V}$ .
2. Cambiar el criterio de asignación de los valores lógicos "0" y "1" a valores de tensión haciendo:

"1" lógico (alta)  $\Rightarrow 0\text{ V}$

"0" lógico (baja)  $\Rightarrow -12\text{ V}$

En estas condiciones, es fácil comprobar que el circuito se comporta como una puerta NAND (figura 3.9).



$A$	$B$	$v_o$ (NAND)
$-12\text{ V}$ ("0")	$-12\text{ V}$ ("0")	$0\text{ V}$ ("1")
$-12\text{ V}$ ("0")	$0\text{ V}$ ("1")	$0\text{ V}$ ("1")
$0\text{ V}$ ("1")	$-12\text{ V}$ ("0")	$0\text{ V}$ ("1")
$0\text{ V}$ ("1")	$0\text{ V}$ ("1")	$-12\text{ V}$ ("0")

**Figura 3.9.** Puerta NAND en RTL

### 3.4. LÓGICA DIODO-TRANSISTOR (DTL)

Al realizar la lógica con diodos y usar al transistor como sumador se obtiene mejor “*fan-out*” y mejor *margen de ruido*. Este tipo de circuito es la base de la familia TTL que comentaremos a continuación. Su interés es puramente académico.

La *figura 3.10* muestra el circuito correspondiente a una puerta NAND de dos entradas. La función lógica la realizan los diodos  $D_1$  y  $D_2$  y el transistor de salida,  $T_1$ , se utiliza para amplificación e inversión. Recordemos que  $V_{CE(sat)} = 0,2 \text{ V}$ ,  $V_{BE(sat)} = 0,8 \text{ V}$  y  $V_{D(on)} \approx 0,7 \text{ V}$  y veamos primero el funcionamiento cualitativo.

La función NAND debe proporcionar, par lógica positiva, una salida en alta ( $v_o = “1” = 5 \text{ V}$ ) ante (00, 01 y 10) y una salida en baja ( $v_o = “0” = 0,2 \text{ V}$ ) ante (11). Veamos que es así.

Para las tres configuraciones (00, 01 ó 10) siempre hay alguna entrada que está apoyada a  $0,2 \text{ V}$ , por consiguiente la tensión en el punto P será:

$$V_P = 0,2 + V_D = 0,2 + 0,7 = 0,9 \text{ V.} \quad [3.4]$$

donde hemos supuesto que  $0,7 \text{ V}$  es la caída de potencial en el diodo cuando este conduce. El circuito equivalente ahora es el de la *figura 3.10.b*.

Para que el transistor conduzca hace falta que en el punto P haya una tensión suficiente para hacer conducir a los dos diodos  $D_3$  y  $D_4$  y que todavía quede tensión suficiente para que el transistor pase a zona activa y conduzca. Así, la  $V_P$  necesaria para que el transistor conduzca será:

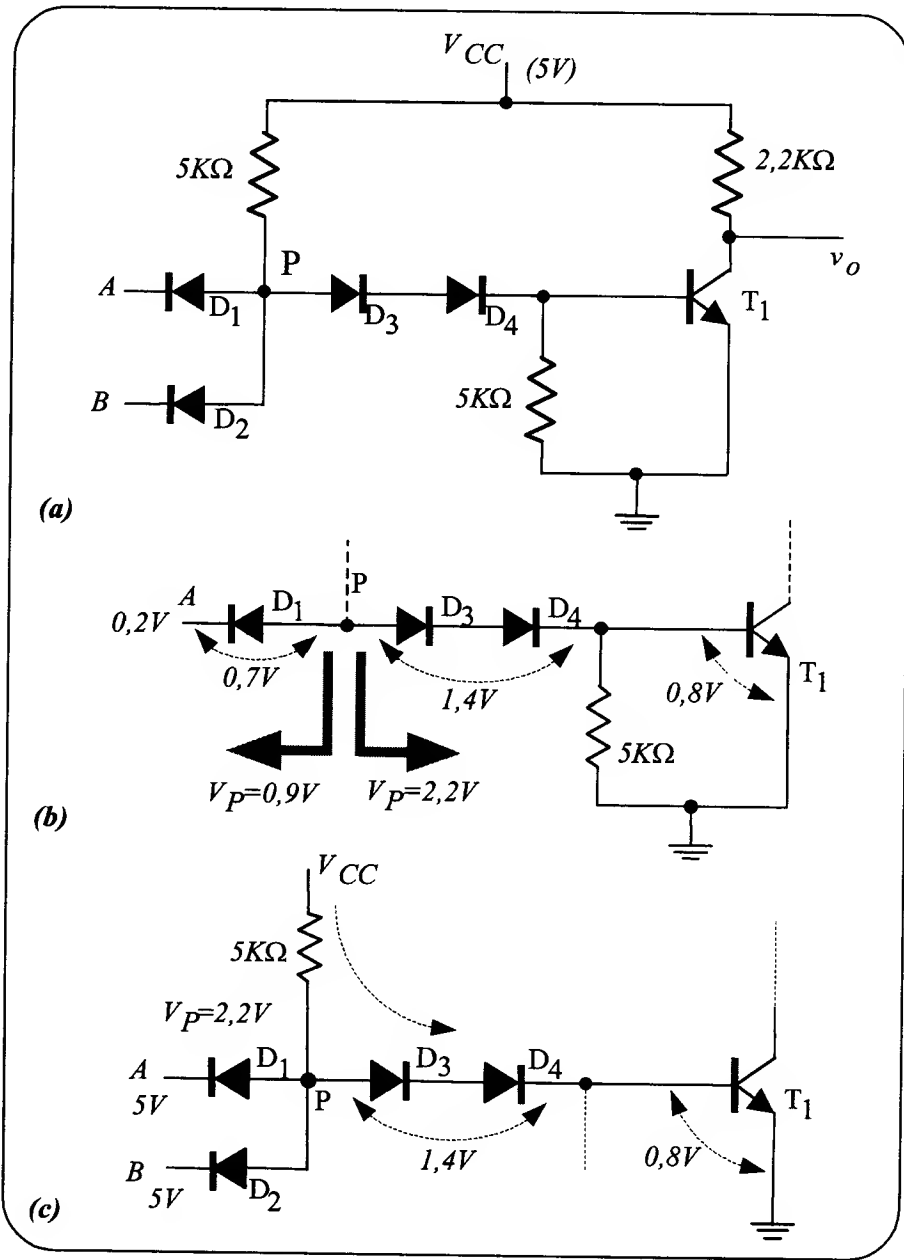
$$V_P = 2 \cdot 0,7 + V_{BEsat} = 1,4 + 0,8 = 2,2 \text{ V} \quad [3.5]$$

Como tenemos  $V_P = 0,9 \text{ V} < 2,2 \text{ V}$ ,  $T_1$  no conduce y  $v_o \approx 5 \text{ V}$ .

Veamos ahora la respuesta a la configuración (11) para comprobar que funciona como una puerta NAND. Si las dos entradas están en alta ( $5 \text{ V}$ ) aparece el circuito equivalente de la *figura 3.10.c*. Los diodos  $D_1$  y  $D_2$  no conducen pero sí conducen en cambio  $D_3$  y  $D_4$  y el valor de  $V_P$  es,

$$V_P = V(D_3) + V(D_4) + V_{BEsat} = 0,7 + 0,7 + 0,8 = 2,2 \text{ V.} \quad [3.6]$$

Esto significa que los diodos de entrada se encuentran con una polarización inversa de  $5 - 2,2 = 2,8 \text{ V}$  justificando nuestra suposición inicial de que estaban al corte.

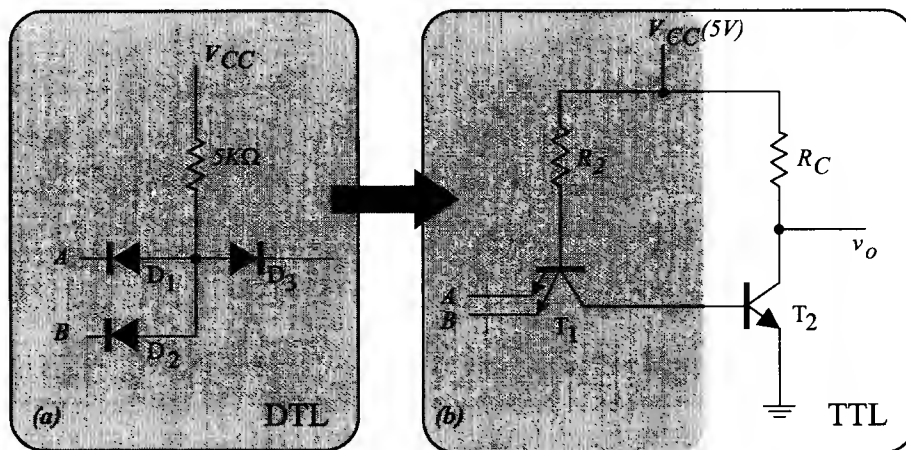


**Figura. 3.10.** Resumen del comportamiento de la puerta NAND en DTL para dos entradas y lógica positiva. (a) Circuito básico. (b) Circuito equivalente para (00, 01, 10) y cálculo de la  $V_P$  necesaria para que  $T_1$  conduzca. (c) Circuito equivalente para la configuración (11).



### 3.5. LÓGICA TRANSISTOR-TRANSISTOR (TTL).

Esta es la familia lógica más importante y de uso actual en tecnología bipolar por su *alta velocidad*, *alto fan-out* y *alta flexibilidad lógica*. El funcionamiento de la puerta básica (NAND) es análogo al de la correspondiente DTL, sustituyendo los diodos de entrada por un transistor multiemisor cuya unión de colector sustituye a su vez a los diodos de umbral  $D_3$  y  $D_4$ . Se añade además una nueva etapa de salida en forma de par activo (totem-pole) que actúa como una fuente de intensidad aumentando el fan-out y la velocidad. El origen de la familia TTL fue intentar minimizar la capacidad parásita ( $C_{cs}$ , colector-sustrato) de la unión N-P de  $T_2$ . Esta unión está polarizada en sentido inverso y por consiguiente su capacidad de transición será tanto menor cuanto menor sea el área de la unión. La *figura 3.11* ilustra el paso de la estructura DTL a la TTL.



**Figura. 3.11.** Paso de DTL a TTL.

El transistor multiemisor  $T_1$  sustituye las funciones de los diodos de lógica y umbral en DTL y presenta un camino de baja impedancia para extraer la carga de  $T_2$  hacia tierra en el paso de saturación a corte, aumentando así la velocidad de conmutación. En resumen el transistor multiemisor sustituye la función del nudo de diodos de entrada haciendo además un camino de baja impedancia desde la base de  $T_2$  a tierra. En esta acción transistor  $T_1$  extrae rápidamente la carga almacenada en la base del transistor  $T_2$  haciendo que TTL sea la familia saturada más rápida.

La otra modificación de DTL que configura la topología típica de la familia TTL es la inclusión

de una etapa activa en cascada a la salida que permite la descarga de  $C_L$  por el transistor en zona activa y la carga de  $C_L$  por el transistor en saturación, como veremos más adelante.

Al ser esta familia un estándar de uso muy frecuente la estudiaremos con más profundidad analizando la puerta NAND básica, viendo otras puertas, comentando las mejoras introducidas sobre el diseño de base (Schottky y Schottky de baja potencia) y terminando el tema con datos de catálogo.

### 3.5.1. Puerta NAND en TTL.

La *figura 3.12.a* muestra el circuito NAND básico de dos entradas y la *figura 3.12.b* el esquema parcial para el análisis de la transición ante el cambio en las configuraciones de entrada. Cualitativamente podemos considerar que los emisores múltiples de  $T_1$  y su unión colector-base sustituyen a los diodos de entrada de DTL y a los de fijación de umbral ( $D_1, D_2$ ).

Si al menos una entrada está en baja,  $V(0) = 0,2 V$ , entonces  $V(P) = 0,2 + 0,7 = 0,9 V$  una de las uniones base-emisor conduce y fija la tensión en P. Para que la unión de colector de  $T_1$  esté polarizada en sentido directo y  $T_2$  pase a conducción se necesitaría del orden de  $3 \cdot 0,7 = 2,1 V$ . Como sólo hay  $0,9 V$ ,  $T_2$  y  $T_3$  están al corte y la tensión de salida sube hacia el valor de la fuente de alimentación:

$$v_O \equiv V("1") \approx 5 V. \quad [3.7]$$

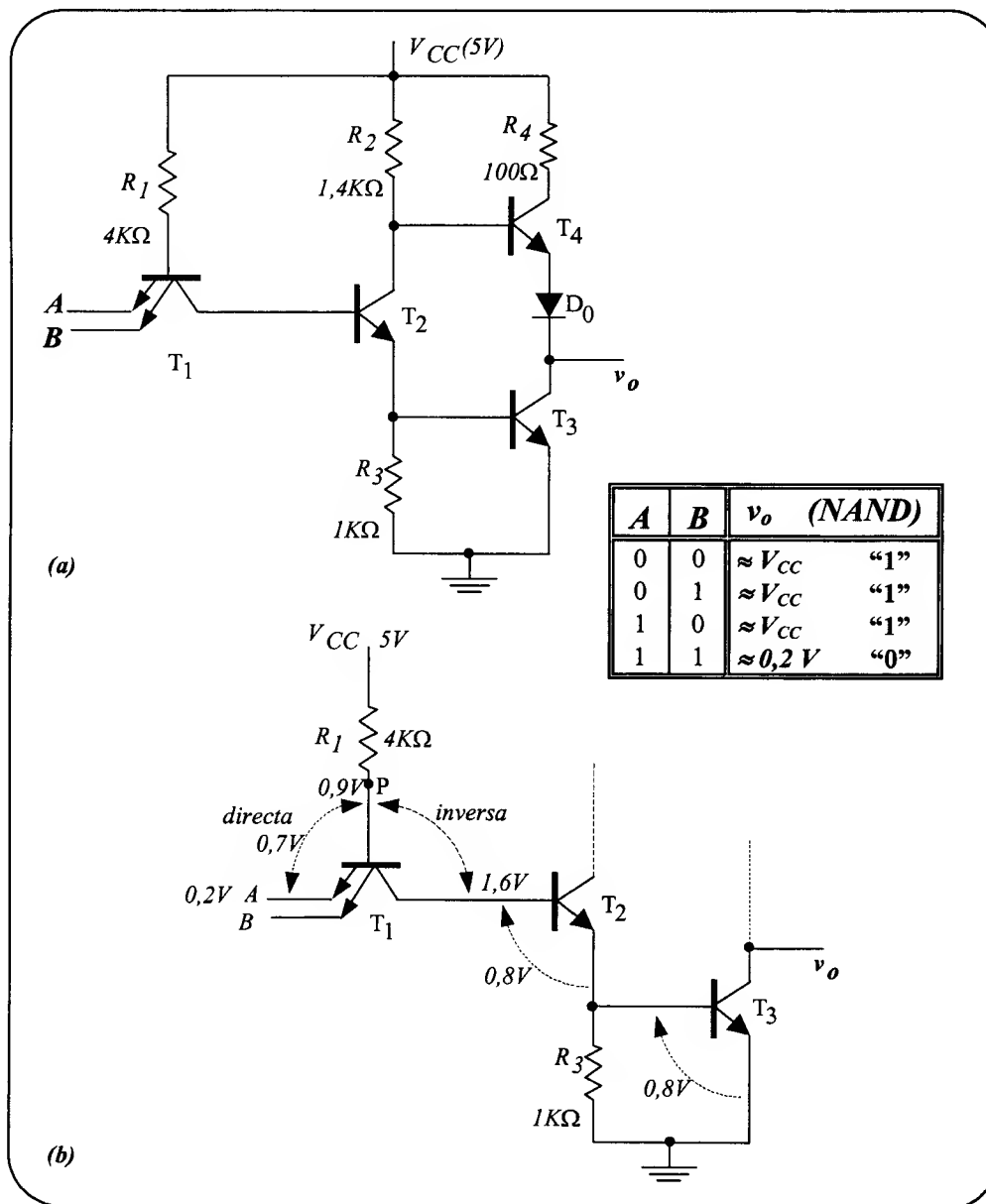
Si todas las entradas están en alta,  $V = 5 V$ , los diodos base-emisor de  $T_1$  están al corte, polarizados en sentido inverso y  $V(P)$  tiende hacia  $5 V$ . Por consiguiente,  $T_2$  y  $T_3$  están en saturación y aparece a la salida

$$v_O = V_{CE sat} = 0,2 V. \equiv V("0") \quad [3.8]$$

y  $V_P$  queda sujeta a  $2,3 V$ .

$$V_P = V_{BE sat}^{T_3} + V_{BE sat}^{T_2} + V_{CB}^{T_1} (V_\gamma \text{ de un diodo}) = 0,8 + 0,8 + 0,7 = 2,3 V \quad [3.9]$$

Tal como hemos explicado el funcionamiento parece que  $T_1$  actúa como dos diodos opuestos y no como un transistor. Sin embargo durante el paso al corte,  $T_1$  actúa como un transistor y reduce el tiempo de almacenamiento necesario para extraer el exceso de carga en la base de  $T_2$  que estaba en saturación. Por otra parte la capacidad parásita colector-sustrato de  $T_1$  está cargada a un cierto potencial.



**Figura 3.12.** Puerta NAND en TTL. (a) Circuito básico. (b) Análisis de la transición.

Si hacemos una transición de alta a baja en cualquiera de los emisores de  $T_1$ , la unión base-emisor de  $T_1$  queda polarizada en sentido directo, de forma que  $T_1$  queda polarizado en zona activa

directa ya que (figura 3.12.b):

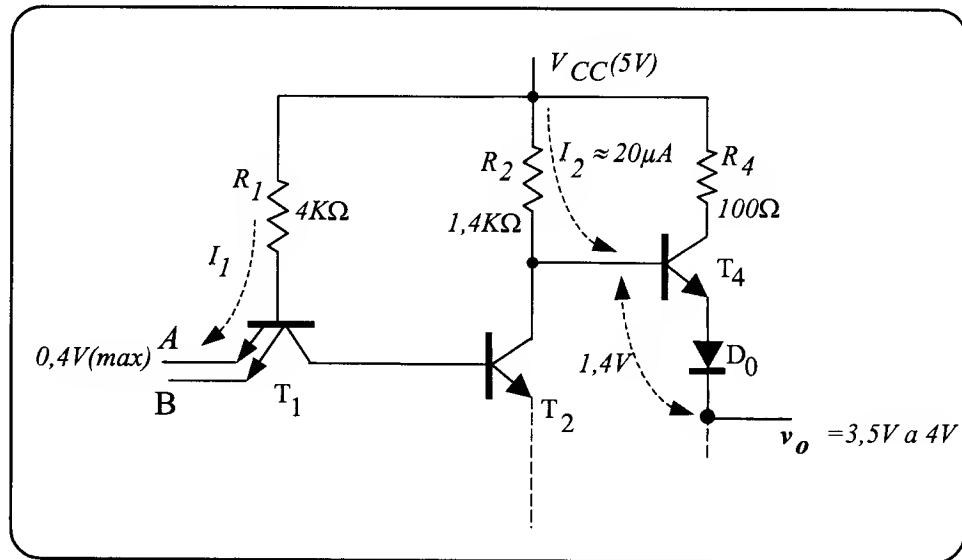
$$V_B(T_2) = 0,8 + 0,8 = 1,6 = V_C(T_1) \quad [3.10]$$

$$V_B(T_1) = 0,2 + V_{BE} = 0,9 \quad [3.11]$$

Esta acción transistor de  $T_1$  provoca un paso considerable de corriente por  $I_{C1}$  que descarga rápidamente la capacidad parásita  $C_{CS}$  y la base de  $T_2$ , pasándolo a corte en poco tiempo. Alcanzada esta situación,  $T_1$  permanece conduciendo una pequeña corriente (la inversa de  $T_2$ ). En estas condiciones, la distribución de corrientes es la que se muestra en la figura 3.13 y la salida es

$$v_o = 5 - I_2 R_2 - V_{BE_4} - V_{D_0} \quad [3.12]$$

dando en general un valor comprendido entre 3,5 y 4 V.

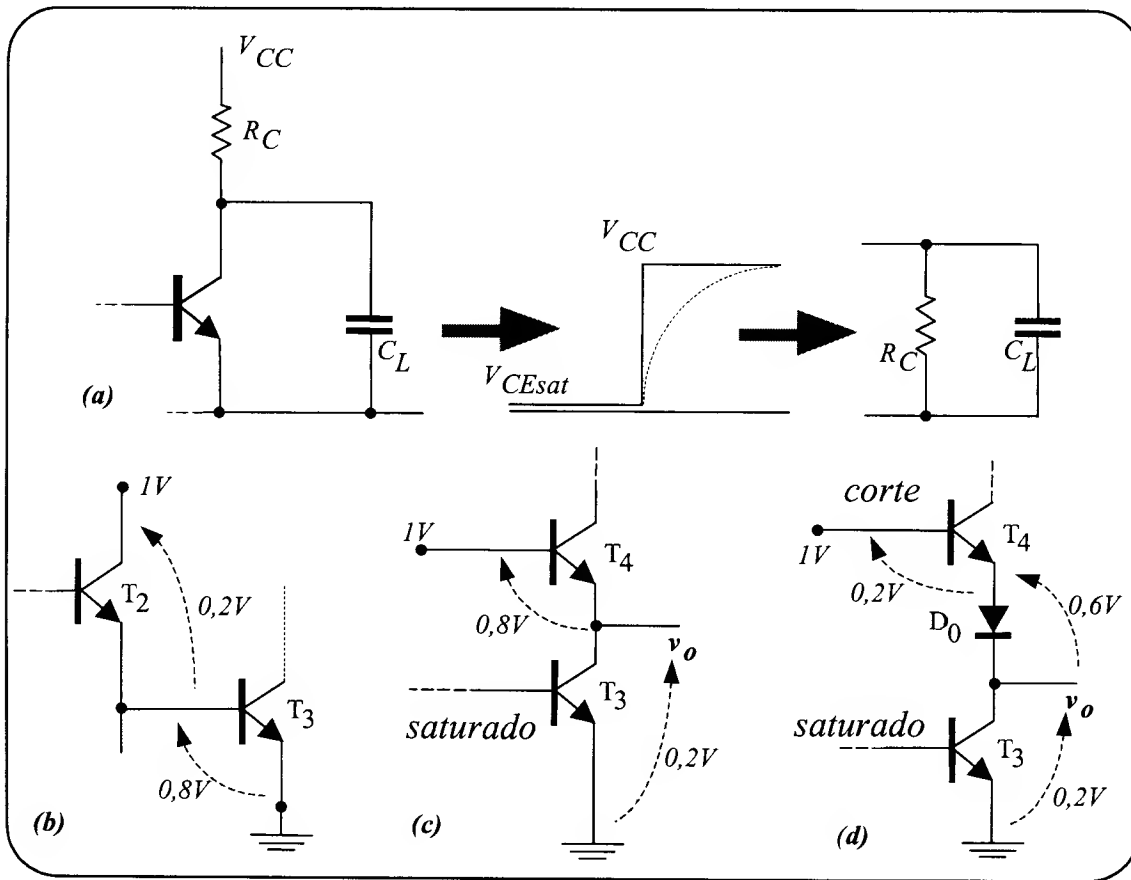


**Figura 3.13.** Puerta NAND en TTL. Distribución de corrientes al final de la transición.

Hemos visto la acción de  $T_1$ . Sigamos adelante viendo cómo actúa  $T_2$ . Este transistor actúa como amplificador saturable, en configuración parafase. Suministra corriente alternativamente a las bases de  $T_3$  y  $T_4$  que forman la etapa de salida.

Veamos por qué se diseña así esta etapa. En la salida de DTL y TTL existe una carga

capacitiva,  $C_L$ , debida a las capacidades de transición de los diodos de entrada en la puerta siguiente, polarizados en sentido inverso. Si tuviéramos una salida por  $R_C$ , cuando la puerta conmuta de baja a alta, el condensador se carga desde  $V_{CEsat}$  hasta  $V_{CC} = 5\text{ V}$  exponencialmente con una constante de tiempo  $\tau = R_C \cdot C_L$  (figura 3.14.a). Así, disminuir el retardo exige disminuir  $R_C$ , pero esto aumenta la disipación de potencia en saturación.



**Figura 3.14.** Detalles de justificación del diseño de la etapa de salida en TTL. (a) Ilustración de la carga del condensador,  $C_L$ . (b)  $T_2$  y  $T_3$  están en saturación cuando la salida está en baja. (c)  $T_4$  y  $T_3$  sin  $D_0$ .  $T_4$  debería estar al corte pero no lo está. (d) Ahora, al introducir  $D_0$ ,  $T_4$  ya está al corte.

La solución TTL es sustituir  $R_C$  por un transistor  $T_4$ , que forma un par activo con  $T_3$ . En la base de  $T_3$  se añade una resistencia,  $R_3$ , para retirar la carga almacenada en la base de  $T_3$  cuando este

tiene que pasar de saturación a corte.  $T_4$  actúa como seguidor de emisor generando una baja impedancia de salida. El diodo  $D_0$  evita que  $T_4$  conduzca cuando  $T_3$  está en saturación, ofreciéndole una resistencia de colector alta.

En la transición de  $T_3$  de saturación a corte, existe un intervalo en el que  $T_3$  y  $T_4$  conducen, ofreciendo un camino de baja impedancia de  $V_{CC}$  a tierra, y un pico de corriente que lo evita  $D_0$ , minimizando así el consumo.

Pasemos a analizarlo de forma cualitativa. La salida está en baja cuando  $T_2$  y  $T_3$  están en saturación (*figura 3.14.b*). En este caso,  $T_4$  debería estar al corte. Veamos que sin  $D_0$ , no lo está. Analizando 3.14.c, tenemos que:

$$V_{BE}^{T_4} = 1 - 0,2 = 0,8 V \quad [3.13]$$

Por lo que debería estar en saturación y no en corte. Veamos cómo al introducir  $D_0$ , se resuelve este problema debido a la caída de  $0,6 V$  que provoca, tal como se ilustra en la *figura 3.14.d*.

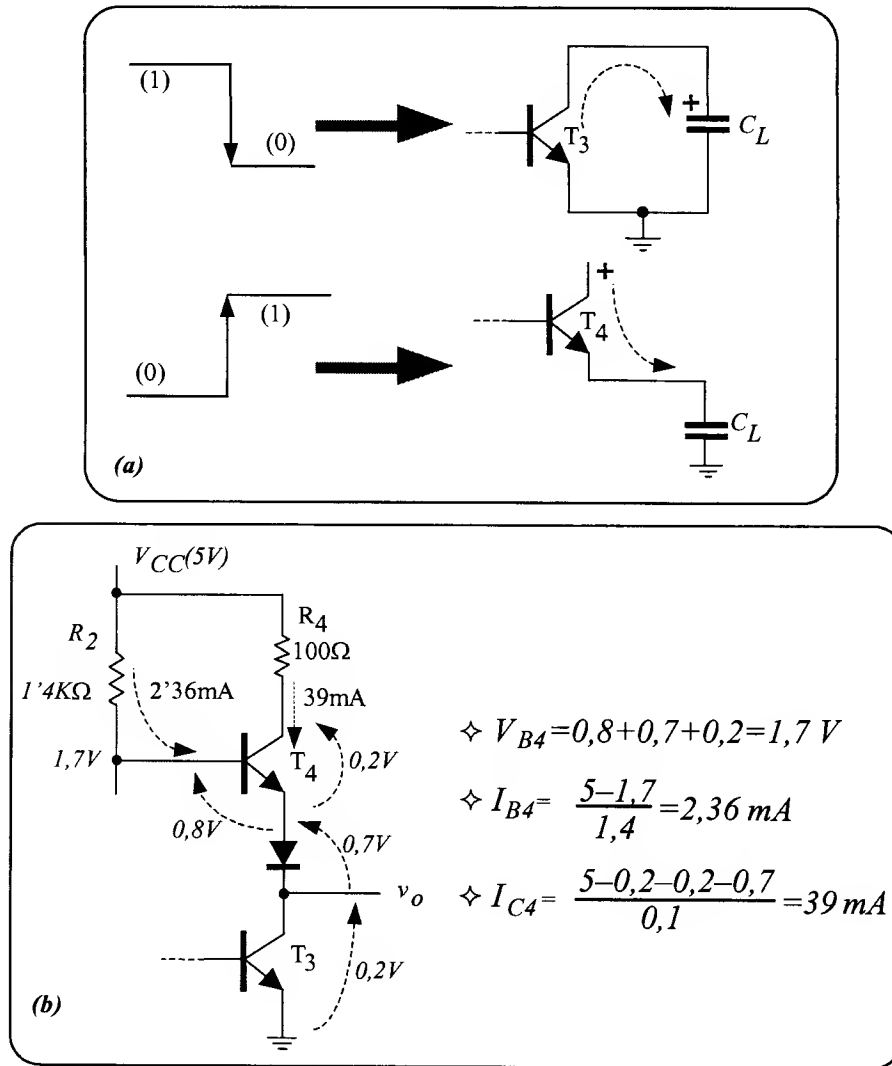
Obsérvese que  $D_0$  también influye disminuyendo el consumo. Ahora, si estamos en alta a la salida,  $C_L$  está cargado a  $V("1")$ . Si excitamos la puerta,  $T_4$  y  $D_0$  van al corte y  $T_3$  conduce y descarga rápidamente  $C_L$  y en la otra transición se carga de nuevo, tal como se ilustra en la *figura 3.15.a*.

En esta segunda transición, se parte de  $T_3$  en saturación (*figura 3.15.b*) y se calculan las corrientes de base y colector en  $T_4$  [ $I_B(T_4) = 2,3 mA$ ] e [ $I_C(T_4) = 39 mA$ ] con lo que podemos calcular el valor de la ganancia mínima necesaria para que  $T_4$  esté en saturación ( $h_{FE \min} = I_C/I_B = 39/2,36 = 16,5$ ). Como la ganancia real suele ser  $\geq 7,5$ , podemos asegurar que  $T_4$  está en saturación, como habíamos supuesto, y carga a  $C_L$  a corriente constante hacia la fuente de alimentación,  $V_{CC} = 5 V$  con una constante de tiempo  $\tau$ ,

$$\tau = (100 + R_{CS}^{T_4} + R_{FUGAS}) \cdot C_L \quad [3.14]$$

pero al aumentar  $v_o$ , disminuye  $I_{C4}$ ,  $T_4$  sale de saturación y  $v_o$  alcanza un valor estacionario. Así pues el valor de la tensión de salida en este caso será:

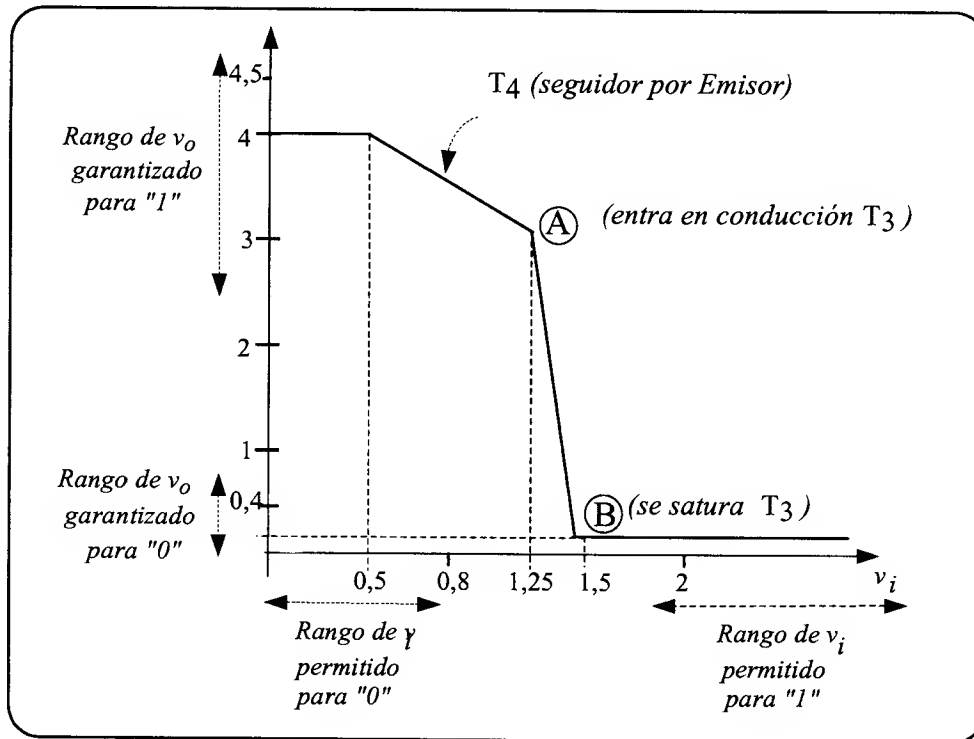
$$v_o = 5 - V_{BE}^{T_4} - V_{D_0} = 5 - 0,5 - 0,6 \approx 3,9 V = V("1") \quad [3.15]$$



**Figura. 3.15.** Configuración del análisis de la etapa de salida. (a) Carga y descarga de  $C_L$  en las transiciones. (b) Análisis de la transición, partiendo de  $T_3$  en saturación.

La figura 3.16 muestra la característica de transferencia de esta familia. Ya conocemos  $v("0")$  y  $v("1")$ . Veamos ahora la transición. El estado de alta permanece prácticamente hasta que  $v_i \approx 0.4$  ó  $0.5V$  y  $V_{B1} \approx 1.1$  ó  $1.2V$  y  $T_1$  entra en zona activa con  $T_2$  que presenta una ganancia controlada por  $R_2/R_3$ . Como  $T_4$  sigue en conducción, la salida es por seguidor por emisor y sigue a la tensión en el colector de  $T_2$  (hasta el punto A), donde entra en conducción  $T_3$ .

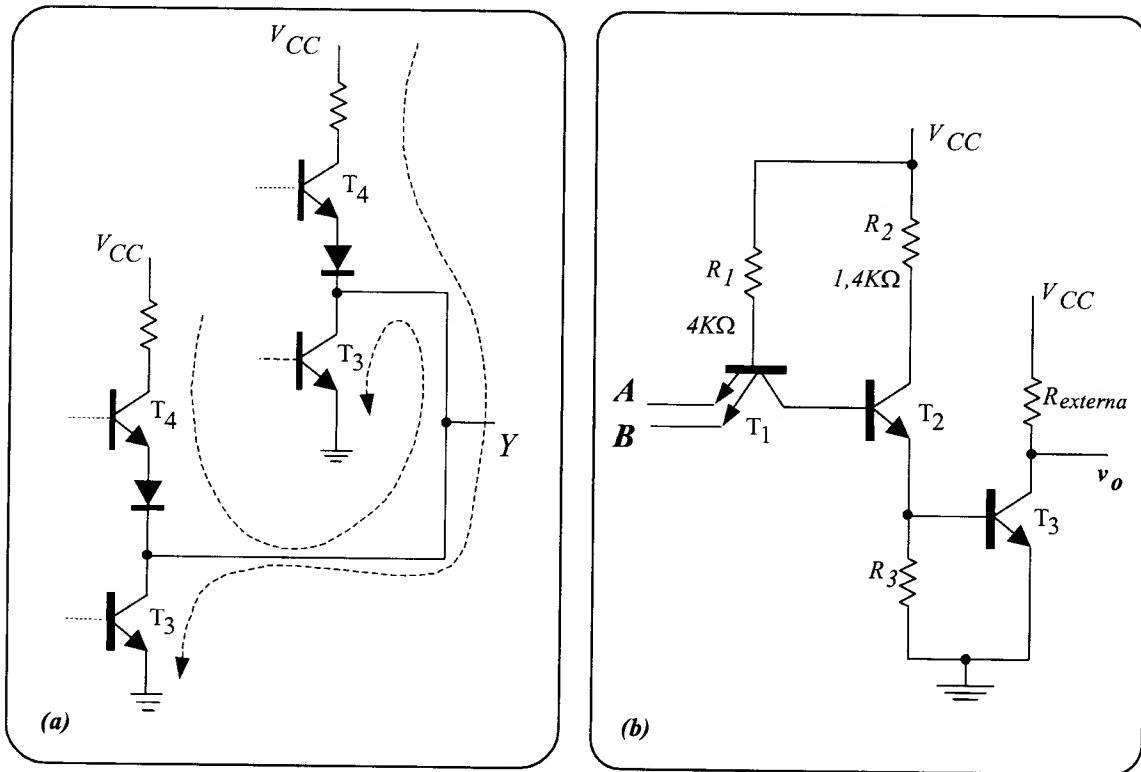
Para  $v \approx 1,4 V$  ya se saturan  $T_2$  y  $T_3$  y el circuito NAND alcanza el nuevo estacionario.



**Figura. 3.16.** Característica de transferencia

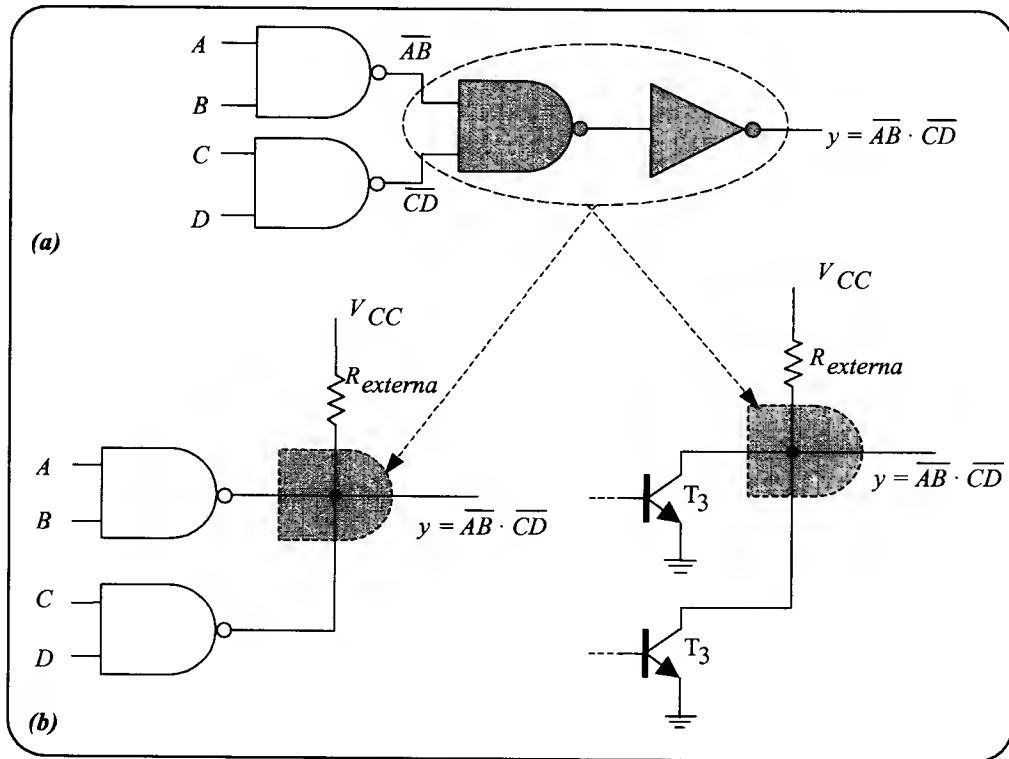
Sobre esta configuración concreta que hemos analizado existen muchas modificaciones específicas cuyo estudio queda totalmente fuera del propósito de este texto. Sólo queremos comentar que la salida por par activo ( $T_3$ ,  $T_4$ ) posee ventajas de *baja impedancia de salida* en ambos estados lógicos y es la responsable del *alto "fan-out"*, *alta velocidad* y *alta inmunidad al ruido*. Sin embargo esta configuración no permite el cableado lógico. Si una salida está en alta y la otra en baja siempre existe un transistor  $T_4$  en saturación ó uno  $T_3$  en conducción, que conectados por el cableado externo producen elevadas corrientes en ambos transistores que pueden destruirlos. Además esta corriente sacaría a  $T_4$  de saturación y confundiría a  $v("0")$  con  $v("1")$  (figura 3.17). Por esto nace la solución de realizar puertas con colector abierto y resistencias de apoyo a la fuente externa de alimentación  $V_{CC}$  (series "open colector").





**Figura.3.17.** Colector abierto. **(a)** Necesidad de abrir los colectores y poner resistencias externas si queremos hacer “cableado lógico”, es decir obtener nuevas funciones a través de la conexión directa de circuitos más elementales. **(b)** Solución de colector abierto.

Ahora ya podemos realizar funciones cableadas. Así por ejemplo la función de la *figura 3.18.a* se puede realizar con cableado lógico mediante la conexión que se muestra en la *figura 3.18.b*. La resistencia externa proporciona un camino de carga a las capacidades internas ( $C_{in}$ ) de las puertas que se conectan a la salida,  $Y$ , y su valor es el de la solución de compromiso entre velocidad, fan-out, disipación y ruido. Si el valor de esta resistencia externa es alto el consumo será bajo, pero la impedancia de salida y las constantes de tiempo serán elevadas. Si la resistencia externa se pone pequeña el fan-out disminuye. La serie 74 de circuitos integrados TTL incluye soluciones de colector abierto. Existen también en TTL puertas de tres estados para la entrada o salida de datos a un canal común al que están conectadas distintas áreas de cálculo o de memoria en un circuito digital de las que hablaremos en el siguiente apartado.



**Figura. 3.18.** Colector abierto. (a) Ejemplo de realización de una función ( $\overline{AB} \cdot \overline{CD}$ ) de forma convencional. (b) Realización de la misma función mediante cableado lógico.

Veamos algunas conclusiones sobre TTL. Su único inconveniente es la generación de ruido por los picos de corriente que se generan durante las transiciones de estado de la etapa de salida en par activo. Esto hace necesario el uso de condensadores de desacoplo externos.

Sus principales ventajas son:

1. *Gran flexibilidad lógica.*
2. *Baja impedancia de salida en "0" y "1".*
3. *Buena inmunidad al ruido.*

Hemos visto en el apartado anterior la estructura interna de una puerta NAND en TTL básica. A partir de aquí y desde 1965 se han ido introduciendo variantes y mejoras encaminadas a aumentar la

velocidad y disminuir el consumo y a resolver problemas específicos de diseño. De estas variantes comentaremos después las series en colector abierto y las puertas de tres estados (“0”, “1” y alta impedancia). Vamos a mencionar ahora las variantes Schottky que dan lugar a las soluciones actuales de alta velocidad (serie 74AS) y bajo consumo (serie 74ALS, “Advanced Low Power”). El origen de las series Schottky está en el intento de eliminar el tiempo de almacenamiento asociado a las uniones colector-base de los transistores. Para ello se conecta en paralelo con la unión de colector un diodo de unión metal-semiconductor (diodo Schottky) que es muy rápido debido a que no almacena carga y fija la tensión base-colector a un valor bajo.

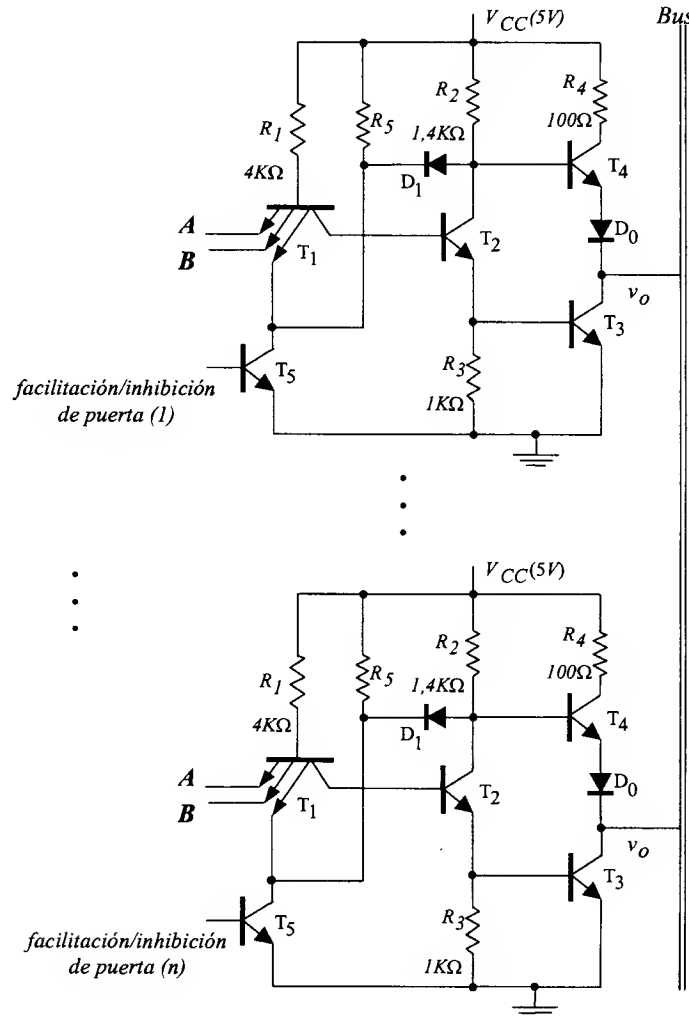
### 3.6. PUERTAS TTL DE TRES ESTADOS

Una forma frecuente y eficaz de conectar distintos circuitos lógicos es el bus (barra de cables conductores compartidos por un conjunto de circuitos) , en donde una señal transmitida por uno cualquiera de los circuitos conectados puede ser recibida por cualquiera de los otros circuitos conectados. Para que esto sea posible electrónicamente debemos dotar a las puertas lógicas de un *tercer estado* de alta impedancia que desconecte funcionalmente a todos los circuitos no seleccionados, aunque están físicamente conectados al bus. Así, cada puerta lógica puede transmitir cada uno de sus dos estados lógicos (“0” ó “1”) cuando está funcionalmente conectada o quedar inhibida, en estado de alta impedancia, mientras accede al bus cualquiera de las otras puertas conectadas.

La *figura 3.19* muestra el circuito de una puerta NAND TTL básica, como la que hemos analizado con detalle previamente (*figura 3.12.a*) modificada adecuadamente para incluir el nuevo estado de alta impedancia. Para ello, hemos incluido un nuevo emisor en el transistor multiemisor  $T_1$  y hemos añadido un nuevo transistor,  $T_5$ , y un diodo  $D_1$ . Cuando la entrada de facilitación en la base de  $T_5$  está en baja el transistor (NPN) no conduce, el diodo  $D_1$  no altera la tensión de colector de  $T_2$  y el circuito de salida funciona de forma normal, pasando a alta cuando en los otros dos emisores del transistor multiemisor  $T_1$  hay 00, 01 ó 10 y pasando a baja ante 11. Es decir, la puerta NAND está física y funcionalmente conectada al bus.

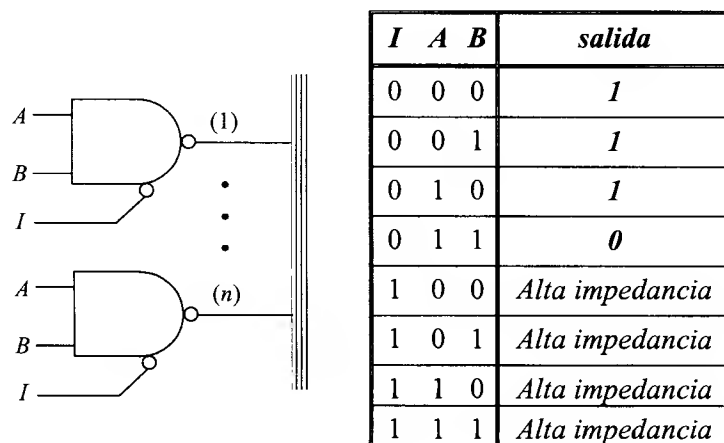
En cambio, cuando la entrada de facilitación está en alta, el transistor  $T_5$  conduce y el tercer emisor de  $T_1$  se fija al valor de saturación de la tensión colector-emisor de  $T_5$ . Por consiguiente, el diodo  $D_1$  conduce y corta a  $T_2$ ,  $T_3$  y  $T_4$ , haciendo que la salida quede aislada tanto de tierra como de

la fuente  $V_{CC}$  por una alta impedancia (tercer estado), independientemente de la configuración lógica que exista en ese momento en los otros emisores ( $A$  y  $B$ ) del transistor multiemisor,  $T_1$ . Evidentemente, en un circuito integrado real el proceso es más complejo, pero la idea del diseño es siempre la misma: Incluir una entrada adicional que permita aislar la salida de fuente y tierra por una alta impedancia de forma que aunque el circuito esté físicamente conectado al bus, pueda no estarlo funcionalmente.



**Figura 3.19.** Puertas TTL de tres estados para conexión a bus.

Esta entrada adicional se representa como se ilustra en la *figura 3.20*, donde se incluye también la tabla de verdad.



**Figura 3.20.** Símbolo lógico y tabla de verdad de las puertas de tres estados.

### 3.7. DATOS DE CATÁLOGO.

Salvo en casos extremadamente sencillos es poco probable que nos baste con algunas puertas lógicas para resolver de forma completa un problema real. Será necesario conectar varios circuitos o buscar funciones integradas más complejas. Dependiendo del grado de integración las funciones se agrupan en cuatro apartados: SSI (integración en pequeña escala), MSI (integración “media escala”), LSI y VLSI (integración en “gran escala” y en “muy gran escala”). En el primer nivel tenemos circuitos con algunas puertas, en el segundo funciones tipo contador o registro y en el tercero y cuarto memorias y circuitos de complicación en el entorno de los microprocesadores.

Para acercarnos al campo de las aplicaciones deben usarse catálogos globales tipo IC-MASTER que organizan los circuitos en términos de su campo de aplicación e independientes del fabricante y, posteriormente, los catálogos del fabricante seleccionado (Texas Instruments, Fairchild, Motorola, AMD, etc...). En digital, los circuitos aparecen ordenados por familias CMOS, ECL, GaAs o TTL y dentro de cada familia por bloques funcionales: funciones aritméticas, inversores, “buffers”, contadores, decodificadores, memorias, multiplexos, osciladores, registros de desplazamiento, procesadores digitales de señal (DSP), etc...

Más adelante comentaremos algunos ejemplos dentro de estos bloques funcionales, casi todos en los niveles MSI y LSI. Ahora vamos a centrarnos en el primer nivel (SSI) donde existen circuitos del tipo:

4 puertas NAND de 2 entradas: SN74AS-00

4 puertas NOR de 2 entradas: SN74AS-02

4 puertas NAND de 2 entradas con salidas en colector abierto: SN74ALS-03B

Las puertas en SSI están organizadas en apartados tales como: Puertas NAND y puertas NOR con y sin colector abierto, expansores, “buffers”, “drivers” y “transceivers”, circuitos biestables, monoestables y multivibradores.

En todos los casos los circuitos participan de los parámetros de la familia, que se ofrecen medidos “en el peor de los casos” y se garantizan dentro del intervalo de temperaturas que caracteriza a la serie. Las series 54ALS y 54AS pueden operar entre  $-55$  y  $125^{\circ}\text{C}$  (rango militar) y las series 74ALS y 74AS entre cero y  $70^{\circ}\text{C}$ .

La velocidad de conmutación se caracteriza por dos parámetros:  $t_{pHL}$  (tiempo de retardo de propagación para el paso de salida en alta a salida en baja) y  $t_{pLH}$  (tiempo de transición de baja a alta). Ambos parámetros se especifican para un pulso de entrada y condiciones de carga usuales (alimentación entre 4,5 y 5,5 V,  $C_L=50$  pF,  $R_L=500$   $\Omega$  y temperatura dentro del rango de la serie). Bajo estas condiciones los tiempos son del orden de 4 ns para ALS y 1,7 ns para AS.

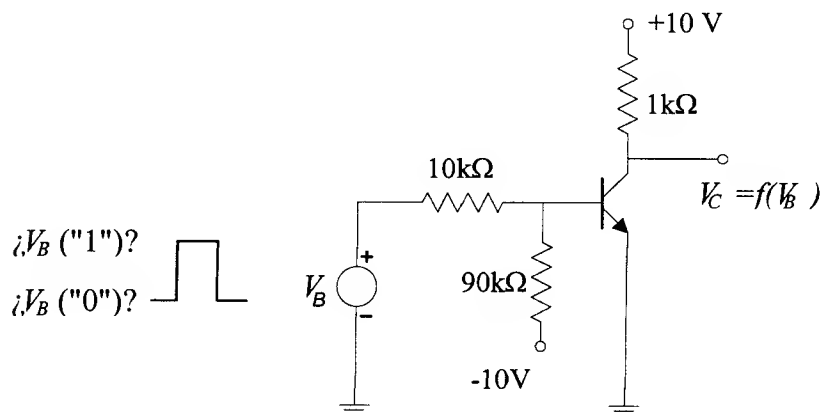
Los márgenes de ruido en continua especifican los incrementos de tensión que todavía garantizan el perfecto funcionamiento del circuito, sin confundir sus niveles lógicos. Como vimos al comienzo del tema, se define para todas las familias como la diferencia entre el valor de la entrada en el peor de los casos ( $V_{IH}$  mínimo ó  $V_{IL}$  máximo) y la salida garantizada también en el peor de los casos ( $V_{OH}$  mínima ó  $V_{OL}$  máxima) especificada para atacar a otras entradas.

La tensión de entrada en estas series puede encontrarse en una de estas tres regiones: *baja* (entre tierra y 0,8 V), *umbral de transición* (entre 0,8 y 2 V) y *alta* (entre 2 V y la alimentación). Si, como consecuencia del ruido, una tensión de entrada realiza excursiones que no la hacen salir de los estados de baja o alta en los que se encontraría en ausencia de ruido, no hay problema. Así, para las familias AS y ALS podemos admitir cambios en baja entre 0,8 V y tierra y en alta entre 2 V y 5,5 V sin que produzcan ningún efecto sobre la salida del circuito. Cuando se realiza una transición sin querer o

no se realiza queriendo, decimos que el circuito ha perdido su función debido al ruido. El ruido en continua no es la única causa de error. La existencia de acoplos capacitivos entre puertas hace necesario considerar en muchas aplicaciones el efecto del ruido dinámico. Su tratamiento sin embargo queda fuera del alcance de este curso.

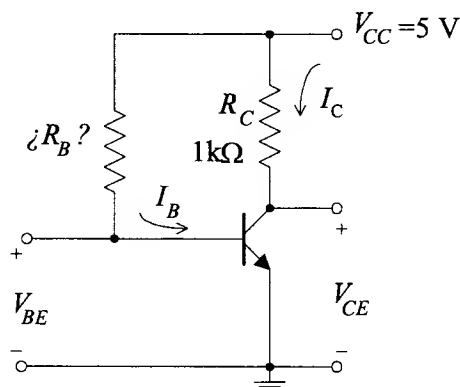
### 3.8. PROBLEMAS (\*)

- E.3.1.** El circuito de la figura adjunta es un sencillo inversor realizado con un transistor bipolar NPN con ganancia en corriente  $\beta_F=100$ , tensión de despegue  $V_{BE(ON)} = 0,5 \text{ V}$  y  $V_{CE(sat)} = 0,2 \text{ V}$ . Construir su característica de transferencia,  $V_C = f(v_B)$ , distinguir las regiones de corte, transición y saturación. ¿Qué rango de valores debe tomar la entrada,  $V_B$ , para que el inversor recorra toda la característica de transferencia?. Es decir, ¿cuál debe ser el valor de  $V_B(0)$  y de  $V_B(1)$  para asegurar la conmutación?.

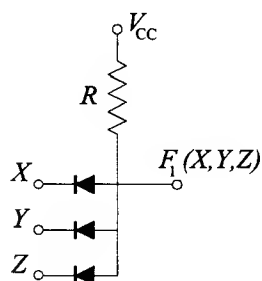


- E.3.2.** El circuito de la figura muestra un inversor polarizado con una sola fuente de 5 V. La ganancia en corriente para continua es  $\beta_F \geq 80$  y  $V_{CE(sat)} = 0,2 \text{ V}$ . ¿Cuál debe ser el valor de  $R_B$  para garantizar la saturación sabiendo que  $R_C = 1k\Omega$  y que  $V_{BE(sat)} = 0,8 \text{ V}$ ?

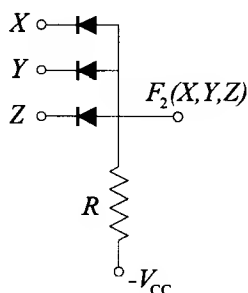
(\*) La solución de estos ejercicios se encuentra en el texto de problemas.



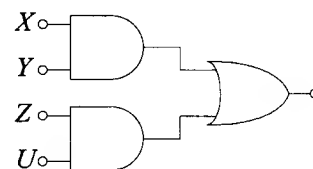
**E.3.3.** En DTL la lógica AND y OR la realizan los diodos. ¿Qué función realizan los sencillos circuitos de las figuras 3.3.a y 3.3.b?. ¿Cómo podrías realizar, usando sólo diodos y resistencias, la función de la figura 3.3.c?. En los tres casos elegir un valor para la alimentación,  $V_{CC}$  y para los niveles lógicos de las variables  $X$ ,  $Y$ ,  $Z$  y  $U$ .



(a)



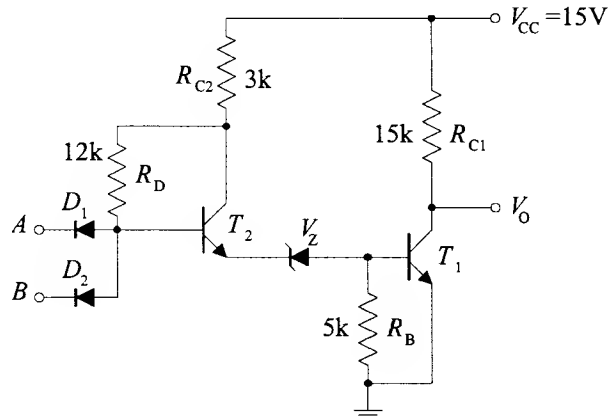
(b)



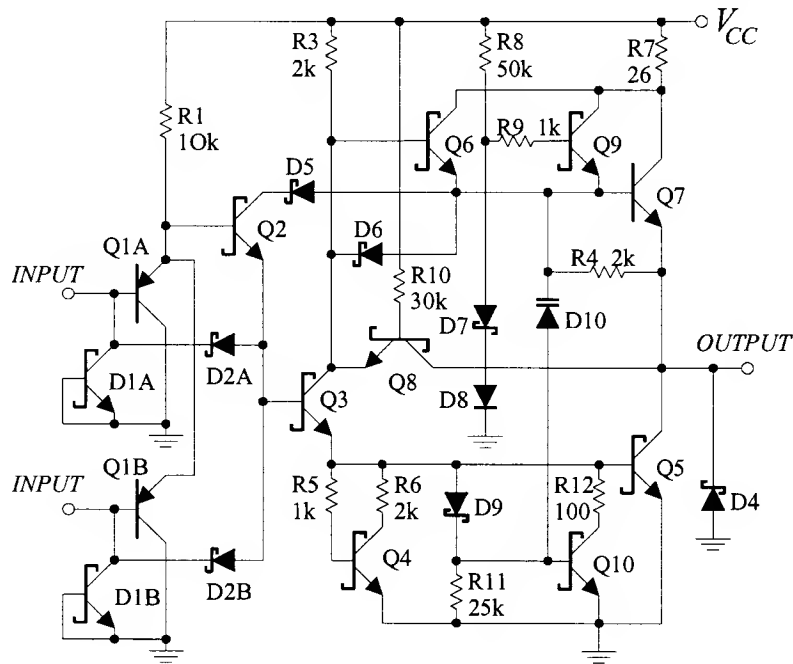
(c)

**E.3.4.** El circuito de la figura muestra una puerta HTL (High-Threshold-Logic) que se usa en aplicaciones industriales por su alta inmunidad al ruido. Se trata de una modificación de DTL en la que el diodo intermedio ha sido sustituido por un zener de 7 V. Analizar el circuito, obtener su característica de transferencia y estimar los márgenes de ruido en el nivel "0" y en el "1".





**E.3.5.** El circuito de la figura adjunta muestra el esquema de un puerta NAND en TTL Schottky (AS00). Sigue el camino de la señal desde la entrada hasta la salida para la configuración (00) y cualquiera de las otras (01, por ejemplo), comparando con el análisis seguido en el tema para la puerta NAND TTL básica, resumido en las figuras 3.12, 3.13. y 3.14.



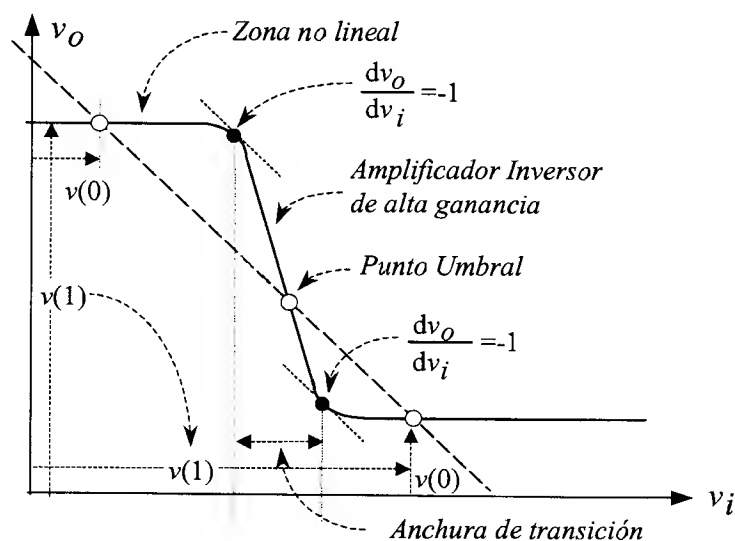
## Preparación de la Evaluación

### Objetivos

- 1: Caracterización de Familias Lógicas
- 2: Circuitos RTL y DTL
- 3: Familias Lógicas TTL

### 1. Caracterización de Familias Lógicas

- 1.1. La caracterización de un circuito digital integrado se realiza sólo a partir de medidas en sus terminales externos. ¿Podría describir las tres características estáticas y las medidas necesarias para obtenerlas?
- 1.2. La figura adjunta ilustra la forma de una característica de transferencia para un inversor. Usando esta figura como apoyo, ¿podría definir los siguientes parámetros?: niveles lógicos, salto lógico, punto umbral, puntos de ganancia unidad, ancho de transición.



- 1.3.** *Las puertas lógicas no suelen usarse de forma aislada sino que en general una puerta es excitada por otra idéntica y, a su vez, ellas excitan a otras puertas de la misma familia. ¿Qué restricciones exige la condición de compatibilidad entre entradas y salidas?*

*Recuérdese que el margen de entradas permitidas para el nivel “0”,  $v_i(0)$ , es aquel para el que podemos garantizar que la salida es “1”,  $v_o(1)$ . Análogamente, el margen de entradas permitidas para el nivel “1”,  $v_i(1)$ , es aquel para el que podemos garantizar que la salida está en “0”,  $v_o(0)$ . Por consiguiente, las restricciones sobre las que preguntamos tienen que ver con el margen de seguridad en los valores de  $v_i(0)$  y  $v_i(1)$  y en los de  $v_o(0)$  y  $v_o(1)$  que actuarán como entradas en la etapa siguiente.*

- 1.4.** *¿Podría explicar el significado de los parámetros usados para caracterizar la inmunidad al ruido de un circuito lógico?. ¿Cómo definiría el margen de ruido en “0” y “1” sobre la característica de transferencia?.*

## **2. Circuitos RTL y DTL**

- 2.1.** *¿Podría reproducir el análisis del circuito inversor de la figura del problema E.3.1?. ¿Qué rangos de valores debe tomar la entrada,  $V_B$ , para que el transistor pase de corte a saturación y viceversa?.*

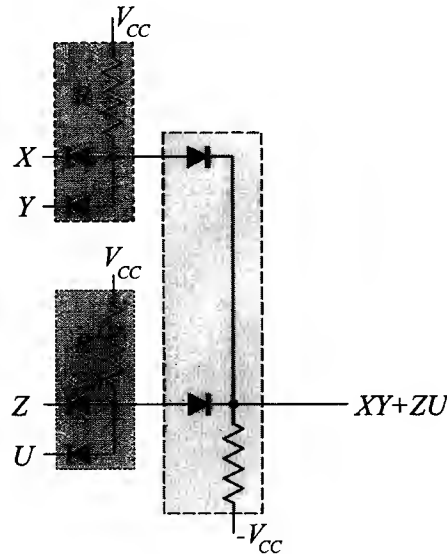
*El problema E.3.1. es representativo de todo el comportamiento del inversor. De hecho incluye todo el proceso de análisis con el cálculo de los valores numéricos en tensiones y corrientes, la obtención de la característica de transferencia y la simulación del funcionamiento del transistor en corte, activa y saturación.*

- 2.2.** *¿Podría proponer un inversor con una sola fuente de continua ( $V_{CC} = 5\text{ V}$ )?. ¿Cuáles serían entonces los valores de  $V(“0”)$  y  $V(“1”)$ ?.  
**2.3.** *¿Podría explicar el funcionamiento de la lógica DTL basándose en el funcionamiento del circuito de la figura adjunta?.**

*Le recordamos que si tiene dificultades en la contestación de esta cuestión puede revisar los ejercicios de evaluación del tema 2 y la solución del problema E.3.3 del texto de problemas.*

*Obsérvese que el circuito tiene dos niveles y que en el segundo nivel los diodos están conectados a la alimentación en sentido opuesto. Téngase en cuenta también que en el primer nivel*

de lógica el ánodo de los diodos está conectado a una tensión positiva ( $+V_{CC}$ ) y en el segundo nivel el cátodo está conectado a una tensión negativa ( $-V_{CC}$ ).



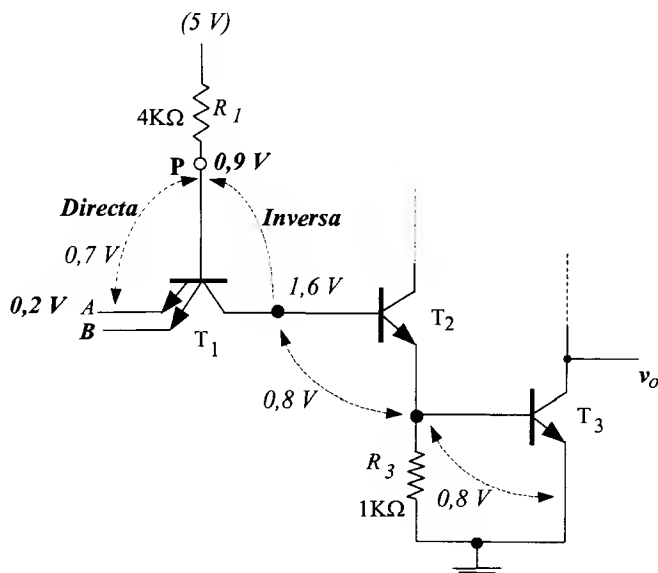
- 2.4. Hemos visto la síntesis de operaciones AND y OR con diodos. ¿Podríamos obtener puertas NAND y NOR usando sólo diodos y resistencias? Intente justificar la respuesta.

### 3. Familia Lógica TTL

- 3.1. Recordando el comportamiento de los circuitos DTL y el circuito correspondiente a una puerta NAND en TTL, tal como la que aparece en la figura 3.12.a de este tema. ¿Podría distinguir qué parte del circuito se encarga de sintetizar las dos subfunciones necesarias para la síntesis de la función NAND. Es decir, ¿qué parte del circuito realiza la función AND y qué parte se encarga después de negarla?.
- 3.2. Para comprobar su nivel de comprensión del funcionamiento de las puertas NAND en TTL le sugerimos que intente describir cualitativamente el funcionamiento de la puerta NAND de dos entradas de la figura 3.12, redactando frases del tipo: “si A ó B están en baja (saturación de una etapa anterior), tendrá 0,2 V y como para conducir  $T_1$  necesita 0,7 V entre Base y Emisor, en la Base tendremos 0,9 V, etc... por consiguiente  $T_3$  no conduce y en Colector tiene 5 V.

No importa tanto su respuesta concreta como la evidencia de que entiende el mapa de tensiones y corrientes que hace que ante la configuración de entrada 11 la respuesta sea 0 y ante cualquiera de las otras configuraciones, la respuesta sea 1.

- 3.3. ¿Podría enlazar el razonamiento cualitativo que le hemos pedido en el ejercicio anterior con los valores numéricos que le ofrecemos ahora?. Justifique estos valores. Si tiene problemas vuelva al apartado 3.5 de este tema y al texto de problemas (E.3.4 y E.3.5)



- 3.4. Explique el resumen de la tabla adjunta, correspondiente al comportamiento de la puerta NAND Schottky analizada en el problema E.3.5.

A	B	$Q_{1A}$	$Q_{1B}$	$V_P \equiv V_B(Q_2)$
0 ( $V_A \approx 0,2V$ )	0 ( $V_B \approx 0,2V$ )	Conducción	Conducción	$\approx 0,9V \rightarrow 0$
0 ( $V_A \approx 0,2V$ )	1 ( $V_B \approx V_{CC}$ )	Conducción	Corte	$\approx 0,9V \rightarrow 0$
1 ( $V_A \approx V_{CC}$ )	0 ( $V_B \approx 0,2V$ )	Corte	Conducción	$\approx 0,9V \rightarrow 0$
1 ( $V_A \approx V_{CC}$ )	1 ( $V_B \approx V_{CC}$ )	Corte	Corte	$\approx V_{CC} \rightarrow 1$



En la dirección de la red *www.onsemi.com* se encuentran los datos de catálogo de las familias ALS y AS. Es conveniente que ya desde el nivel de puertas el alumno tome contacto con los circuitos integrados reales correspondientes a las estructuras internas que se estudian en la Electrónica Digital. Por consiguiente, le sugerimos que visite estas páginas y observe la estructura de los datos que contienen:

- ✧ Descripción externa del circuito
- ✧ Tabla de verdad
- ✧ Aspecto anecdótico del encapsulado
- ✧ Símbolo lógico
- ✧ Diagrama lógico (para lógica positiva y/o negativa)
- ✧ Características eléctricas (alimentación, niveles de alta y baja, márgenes de ruido, tiempos de conmutación y por consiguiente velocidad, etc...).

## REFERENCIAS BIBLIOGRÁFICAS

1. Harris, J.N., Gray, P.E., Searle, C.L.: “*Digital Transistor Circuits*”. Semiconductor Electronics Education Committee, vol. 6 . John Wiley, 1966.
2. Muñoz, E.: “*Circuitos Electrónicos Digitales II*”. ETSIT. Madrid, 1981
3. Tabú, H. and Schilling, D.: “*Electrónica Digital Integrada*”.. Marcombo, barcelona, 1984.
4. Texas Instruments. “*The TTL Data Book*”. 1987.
5. <http://www.onsemi.com>



# TEMA 4

---

## ***Familias lógicas (II): ECL, MOS, CMOS y BiCMOS***

---

### **+ CONTEXTO**

*Todas las familias lógicas estudiadas en el tema anterior (RTL, DTL y TTL) son de lógica saturada. Es decir, los transistores bipolares operan entre corte y saturación. Esto limita mucho la velocidad de conmutación porque cada vez que un inversor cambia de estado hay que extraer toda la carga almacenada en las regiones neutras y en las zonas de transición de las uniones P-N. De hecho gran parte de los desarrollos tecnológicos que han marcado la evolución de la familia TTL (versiones AS y ASL) han estado asociados a aumentar la velocidad de conmutación sin perjudicar el consumo, usando diodos Schottky no saturados y de baja disipación.*

*En este tema completamos el panorama de las distintas familias lógicas estudiando los*

*circuitos ECL, MOS, CMOS y BiCMOS.*

*La familia ECL (lógica de emisores acoplados) aborda el problema de la alta velocidad de conmutación necesaria en muchas aplicaciones aceptando un consumo mayor y una menor inmunidad al ruido y haciendo trabajar a los transistores bipolares en la zona activa normal. Es decir, sin llegar nunca al corte ni a la saturación. Así, no tienen prácticamente carga almacenada en la zona de transición y la conmutación es muy rápida. La estructura de los circuitos ECL es análoga a la de un amplificador diferencial de alta ganancia.*

*Las familias MOS (Metal-Oxido-Semiconductor) y CMOS (MOS Complementarios de canal N y P) aborda el otro problema que limitaba a la TTL. Su complejidad y, por consiguiente, la baja densidad de integración. Están basadas en los transistores unipolares de efecto campo, donde todos los procesos son superficiales, el consumo es menor y hace falta mucha menos área de silicio para integrar una determinada función. Por otro lado la tecnología CMOS produce dispositivos con mayor aislamiento, mayor velocidad y mayor densidad de integración, lo que la hace idónea para diseñar circuitos estáticos y dinámicos en prácticamente todos los campos de la Electrónica Digital (memorias volátiles y no volátiles, lógica programable, procesadores, etc...).*

*La importancia de los procesos CMOS en el estado actual de la tecnología de integración en muy gran escala (VLSI) y “ultra gran escala (ULSI) es realmente muy alta.*

## **++ CONOCIMIENTO PREVIO NECESARIO**

*Al igual que el conocimiento previo necesario para comprender el funcionamiento de los circuitos TTL era el de los transistores bipolares trabajando en corte y saturación, el conocimiento necesario para este tema es el de los transistores bipolares trabajando en zona activa lineal (para ECL) y el de los transistores MOS con canal N y P trabajando en modo de realce (para MOS y CMOS). Todos estos conocimientos se encuentran en el tema 2.*

## **+++ OBJETIVOS DEL TEMA**

**Objetivo 1:** *Comprender la Lógica de Emisores Acoplados (ECL) (funcionamiento, características de transferencia, niveles lógicos y acoplos con otras familias).*



- Objetivo 2:** *Comprender el funcionamiento del circuito inversor básico en tecnología MOS y con carga activa (funcionamiento, característica de transferencia, niveles lógicos y forma de diseñar las puertas básicas NAND y NOR).*
- Objetivo 3:** *Comprender la lógica CMOS y la BiCMOS (funcionamiento del inversor básico, característica de transferencia, comportamiento dinámico y forma de diseñar las puertas básicas NAND y NOR).*
- Objetivo 4:** *Saber comparar las distintas familias lógicas (TTL, ECL, MOS y CMOS) en función de los criterios estudiados al comienzo del capítulo 3 (velocidad, consumo, facilidad de integración, inmunidad al ruido, etc...).*

### ++++ GUÍA DE ESTUDIO

*El proceso de presentación del material didáctico en los temas 3 y 4 se repite. Los pasos a seguir en cada familia lógica son los siguientes:*

- 1. Proponemos un circuito básico (el inversor, la puerta NAND o la puerta NOR).*
- 2. Explicamos su funcionamiento siguiendo el camino de las señales desde las entradas hasta la salida para cada una de las configuraciones lógicas posibles (00, 01, 10, 11, para el caso de dos entradas).*
- 3. Obtenemos la característica de transferencia por procedimientos analíticos o de razonamiento cualitativo. Así conocemos los niveles, los márgenes de ruido y la pendiente de la zona de transición.*
- 4. Cuando el circuito básico ha sido el inversor, continuamos el estudio de la familia viendo cómo se diseñan otros circuitos. En general, las puertas NAND y NOR o las puertas de transmisión.*
- 5. Finalmente, valoramos la familia de acuerdo con una serie de criterios, tales como velocidad, consumo, márgenes de ruido, facilidad y densidad de integración, etc.*

*Este método será también el que seguiremos en este tema. Los problemas resueltos correspondientes se encuentran en el tema 4 del texto de problemas. Aconsejamos el uso del simulador PSpice u otro equivalente en combinación con el texto de problemas para simular los*

*distintos circuitos. El uso de la versión de evaluación del simulador tiene ciertos inconvenientes asociados al límite en el número máximo de transistores que pueden incluirse en un circuito y a la poca variedad en la selección de dispositivos, que no siempre son los más adecuados para su uso en conmutación. Sin embargo, el carácter libre de este software y el propósito esencialmente pedagógico del material de este tema, hacen esencialmente ventajoso su uso.*



## ***Contenido del Tema***

### **4.1. LÓGICA DE EMISORES ACOPLADOS**

La estructura básica de las puertas ECL es la de un *amplificador diferencial* con una entrada tomada como referencia para fijar el umbral. Esta configuración permite además salidas complementarias (OR y NOR, por ejemplo) a través de dos circuitos en colector común (seguidores por emisor) que toman la señal de los colectores de los transistores del par diferencial. Esto permite reducir las impedancias de salida, aumentar la velocidad de conmutación debido a la ganancia en corriente de las etapas en colector común y obtener niveles de tensión a la salida compatibles con las entradas de otras puertas de la misma familia. Desarrollaremos este apartado en tres puntos. Primero recordaremos la forma básica de un amplificador diferencial, después veremos una versión simplificada del inversor ECL que se obtiene al fijar una de las tensiones de base en un punto intermedio entre  $v(0)$  y  $v(1)$  y, finalmente, veremos cómo se pasa del inversor a las puertas NOR.

#### **4.1.1 Amplificador Diferencial**

El circuito de la *figura 4.1* muestra la configuración básica de un amplificador diferencial. El circuito posee dos entradas,  $v_1(t)$  y  $v_2(t)$  y dos salidas  $v_o^1(t)$  y  $v_o^2(t)$ . Dada la simetría del circuito, al tomar la salida entre los dos colectores, se elimina la parte común y sólo queda la diferencia. Para poner esto de manifiesto, definimos también la parte común,  $v_c$ , y la parte diferencial,  $v_d$ , de las entradas:

$$v_c = \frac{v_1 + v_2}{2} \quad \text{y} \quad v_d = \frac{v_1 - v_2}{2} \quad [4.1]$$

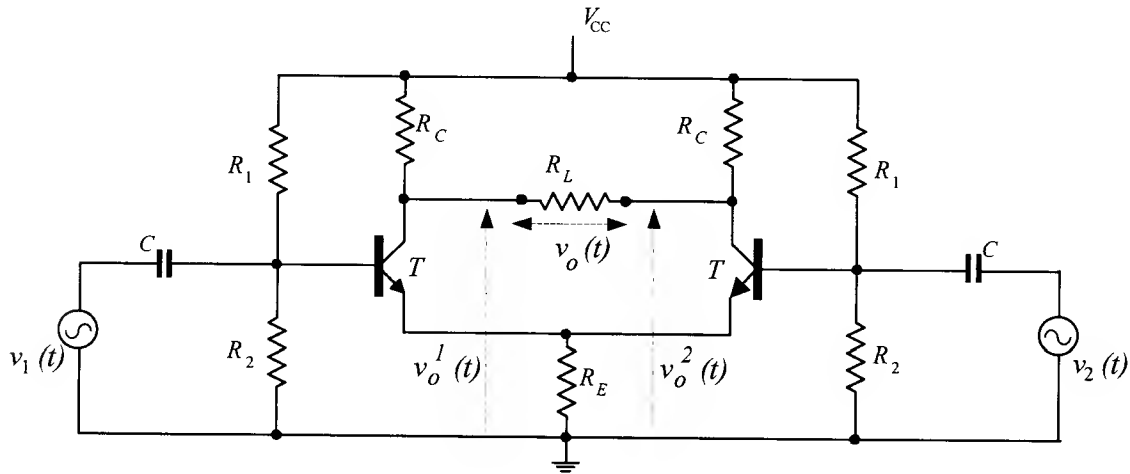
Es decir, siempre podremos escribir  $v_1$  y  $v_2$  como:

$$v_I = v_c + v_d \quad \text{y} \quad v_2 = v_c - v_d \quad [4.2]$$

y, después, aplicar el principio de superposición. Así, cuando sólo tenemos aplicada la entrada  $v_I(t)$ , la salida correspondiente será:

$$v_o^I(t) = A_{cI} \cdot v_c + A_{dI} \cdot v_d \quad [4.3]$$

siendo  $A_{cI}$  la ganancia en “modo común” y  $A_{dI}$  la ganancia en “modo diferencial”.



**Figura 4.1.** Configuración básica de un amplificador diferencial

Análogamente, cuando sólo está presente  $v_2(t)$ , la salida es:

$$v_o^2(t) = A_{c2} \cdot v_c - A_{d2} \cdot v_d \quad [4.4]$$

Por consiguiente, al estar presente las dos entradas,  $v_I(t)$  y  $v_2(t)$ , y tomar la salida flotante, entre los dos colectores, tendremos:

$$v_o(t) = v_o^I(t) - v_o^2(t) = (A_{cI} - A_{c2}) \cdot v_c + (A_{dI} + A_{d2}) \cdot v_d \quad [4.5]$$

Si el circuito es simétrico,

$$A_{cI} = A_{c2} = A_c \quad \text{y} \quad A_{dI} = A_{d2} = A_d \quad [4.6]$$

y por consiguiente, la salida flotante será:

$$v_o(t) = 2A_d \cdot v_d = 2A_d \frac{v_1 - v_2}{2} = A_d(v_1 - v_2) \quad [4.7]$$

Es decir, la salida es proporcional (con coeficiente de proporcionalidad  $A_d$ ) a la diferencia entre las dos señales ( $v_1 - v_2$ ).

Veamos ahora de forma simplificada el cálculo de la ganancia en “*modo común*”, cuando las dos entradas son iguales ( $v_1 = v_2 = v_c$ ) y en “*modo diferencial*”, cuando las dos entradas son opuestas ( $v_1 = -v_2 = v_d$ )

➤ **Ganancia en “modo común”** ( $v_1 = v_2 = v_c$ )

Observemos primero que por la resistencia  $R_E$ , común a los dos emisores, pasa una corriente igual a  $2I_E$ , porque hay dos transistores inyectando corriente. Esta situación, a efectos del cálculo de la tensión en esos emisores ( $V_E$ ) es equivalente a otra en la que pasara  $I_E$  por una resistencia de valor  $2R_E$ ,

$$V_E = (2I_E) \cdot R_E = I_E \cdot (2R_E). \quad [4.8]$$

Esto nos permite usar el circuito simplificado de la figura 4.2.a para calcular la ganancia en modo común, cuya expresión analítica es:

$$A_c = \frac{v_{oc}}{v_c} \approx \frac{i_C R_C}{i_E (r_e + 2R_E)} \approx \frac{R_C}{r_e + 2R_E} \quad [4.9]$$

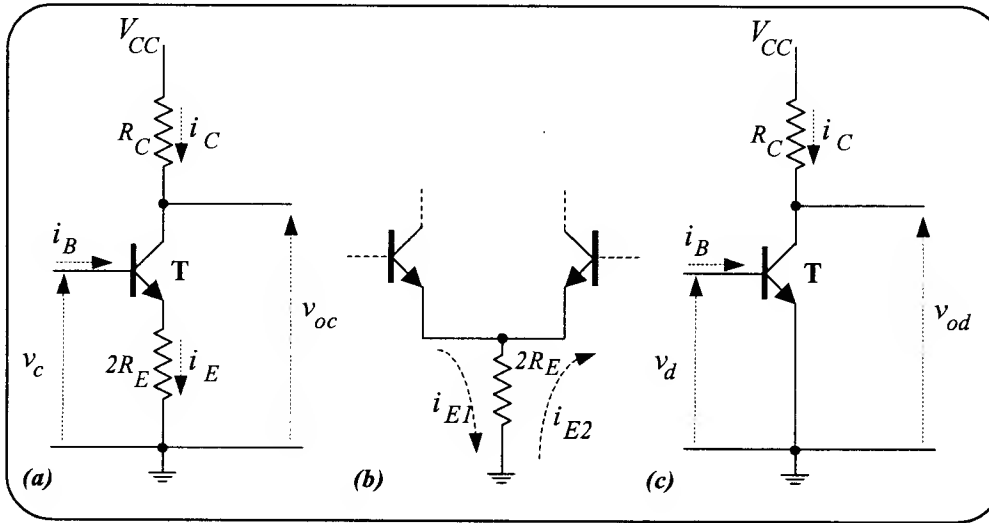
En este cálculo hemos realizado algunas aproximaciones. La primera es suponer que la corriente de colector ( $i_C$ ) y la emisor ( $i_E$ ) son prácticamente iguales. La segunda aproximación ha sido despreciar (como sumando) la caída en la unión base-emisor ( $v_{BE} \approx 0,6$  V). Finalmente, hemos introducido la resistencia interna de emisor del transistor,  $r_e$ .

Este valor de la ganancia en modo común ( $A_c$ ) es *muy pequeño* porque la resistencia de colector,  $R_C$ , es pequeña en comparación con el denominador ( $2R_E + r_e$ ). Además, intencionalmente se diseña haciendo  $R_E$  muy grande mediante un generador de corriente. Finalmente, la resistencia interna del emisor,  $r_e$ , es de unos pocos ohmios y siempre muy inferior a la externa, con lo que todavía podemos simplificar más la expresión de la ganancia,  $A_c$ :

$$A_c \approx \frac{R_C}{2R_E} \quad (\text{con } 2R_E \gg R_C) \quad [4.10]$$

➤ **Ganancia en “modo diferencial”** ( $v_1 = -v_2 = v_d$ )

Veamos ahora la ganancia en modo diferencial. En este caso también podemos simplificar el cálculo observando que al ser iguales y de signo opuesto las tensiones aplicadas en la base de los dos transistores  $T_1$  y  $T_2$ , ( $v_1 = -v_2$ ), las corrientes en sus terminales de base son también de igual magnitud y signo opuesto ( $i_{B1} = -i_{B2}$ ) y lo mismo ocurre en los emisores ( $i_{E1} = -i_{E2}$ ). Por consiguiente, tal como se ilustra en la parte b de la *figura 4.2*, la caída de tensión en la resistencia  $R_E$  es nula por lo que podemos eliminarla y unir el emisor a tierra, tal como se muestra en 4.2.c.



**Figura 4.2.** Ilustración de los circuitos simplificados usados para el cálculo de la ganancia en modo común y en modo diferencial de un amplificador diferencial. **(a)** Modo común. **(b)** y **(c)** Modo diferencial.

Ahora en este circuito simplificado equivalente es donde vamos a calcular la ganancia del modo diferencial,  $A_d$ , repitiendo de nuevo la aproximación de igualar las corrientes de emisor y colector ( $i_C = i_E$ ) y recordando que el valor de la resistencia interna del emisor es  $r_e$ :

$$A_d = \frac{v_{od}}{v_d} \approx \frac{i_C R_C}{i_E r_e} \approx \frac{R_C}{r_e} \quad [4.11]$$

Este valor de  $A_d$  es muy grande porque  $R_C$  es mucho mayor que  $r_e$ . Este gran valor de la ganancia en modo diferencial es el causante de la alta velocidad de conmutación de la familia lógica ECL.

### 4.1.2 Inversor ECL

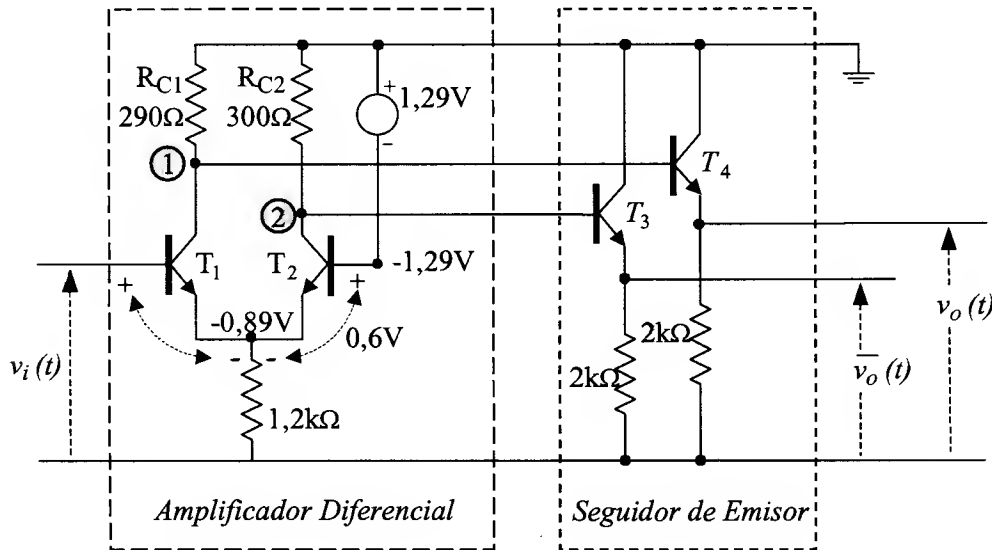
Si cogemos ahora el amplificador diferencial del apartado anterior y fijamos una de las entradas, por ejemplo  $v_2$ , a un valor de tensión intermedio entre los correspondientes a los dos niveles lógicos  $V(0)$  y  $V(1)$ , cada vez que la señal de entrada en  $v_1$  cruza el valor de  $v_2$  el circuito conmuta. Si no dibujamos la parte del amplificador diferencial responsable de la polarización de los transistores  $T_1$  y  $T_2$  y añadimos dos nuevos transistores  $T_3$  y  $T_4$  actuando como seguidores de emisor para apoyar a tierra las señales de los colectores de  $T_1$  y  $T_2$ , obtenemos el circuito del inversor ECL que se muestra en la figura 4.3.

Lo primero que conviene observar es que este circuito tiene dos módulos claramente diferenciados. El primero es un *amplificador diferencial* y el segundo son dos *seguidores de emisor* que ofrecen a la salida  $v_o(t) = v_i(t)$  y  $v_o(t) = -v_i(t)$ .

Lo característico de este amplificador diferencial es que uno de los transistores ( $T_2$ ) tiene como entrada un nivel de tensión constante ( $-1,29\text{ V}$ ) que define el “modo común”. Es decir, cuando en la base del otro transistor ( $T_1$ ) tenemos  $-1,29\text{ V}$  las dos entradas son iguales y la salida diferencial es cero. Este nivel de continua se encuentra entre los valores asociados al “0” lógico y al “1” lógico. Así, al conmutar la entrada de baja a alta (por ejemplo), se atraviesa el valor de referencia y, rápidamente, con toda la ganancia del modo diferencial,  $A_d \approx \frac{R_C}{r_e}$ , el circuito conmuta dando lugar a la *función del inversor*.

La señal entre los colectores de los transistores  $T_1$  y  $T_2$  (puntos 1 y 2) depende de las excursiones del potencial en la base del transistor  $T_1$ , ya que el potencial en la base de  $T_2$  está fijado a un valor de referencia de  $-1,29\text{ V}$ . Recordemos que en este circuito ECL los transistores  $T_1$  y  $T_2$  no pasan de corte a saturación sino que siempre están trabajando en zona activa y, por tanto, siempre conducen, aunque con distintos valores de corriente de colector ( $i_{C1} \neq i_{C2}$ ), que son los que van a dar lugar a distintos valores en la tensión diferencial de salida ( $v_{C1} - v_{C2}$ ), que se traslada después a la base de los transistores seguidores de emisor,  $T_3$  y  $T_4$  para producir las salidas  $v_o(t) = v_i(t)$

$$y \ v_o(t) = -v_i(t).$$



**Figura 4.3.** Inversor ECL

Veamos un ejemplo numérico de cálculo de los niveles de la tensión de entrada,  $v_i$ , correspondientes a los niveles lógicos “0” y “1”. Decimos que la entrada es “0” cuando el potencial en la base de  $T_1$  es algo inferior al potencial en la base de  $T_2$ . Es decir,

$$v_i(\text{“0”}) = -1,29 - \text{“algo”} = -1,29 - 0,31 = -1,6 \text{ V}$$

Por ejemplo

Análogamente, decimos que la entrada es “1” cuando el potencial en la base de  $T_1$  es algo superior al potencial en la base de  $T_2$  (valor de referencia, -1,29V).

$$v_i(\text{“1”}) = -1,29 + \text{“algo”} = -1,29 + 0,29 = -1 \text{ V}$$

Por ejemplo

Estas pequeñas diferencias son suficientes para que la alta ganancia del amplificador diferencial



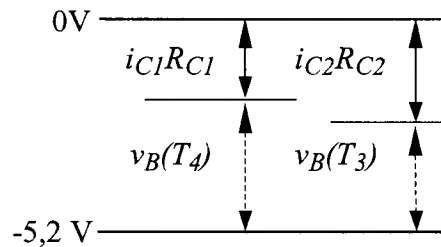
produzca una diferencia entre los valores de las tensiones de colector. Así, en el caso de  $v_i = "0"$ ,  $i_{C1}$  es menor que  $i_{C2}$  y  $v_{C1} = i_{C1} R_{C1}$  está a un potencial superior (en relación con el valor de la tensión de alimentación de los emisores,  $-5,2V$ ) al de  $v_{C2} = i_{C2} R_{C2}$ , ya que  $R_{C1} \approx R_{C2}$ . Esta diferencia de potencial,  $v_{C1} - v_{C2}$ , es la que se transmite a la salida a través de  $T_3$  y  $T_4$ .

Ahora la tensión en la base de  $T_4$  es:

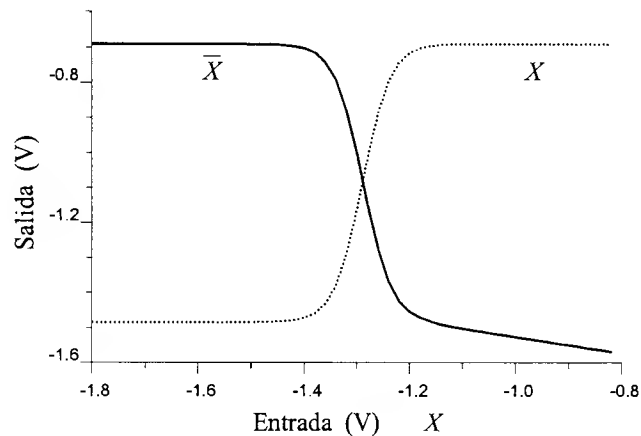
$$v_B(T_4) = v_{C1} = -5,2 + i_{C1} R_{C1}$$

Análogamente

$$v_B(T_3) = v_{C2} = -5,2 + i_{C2} R_{C2}$$



De forma complementaria, en el caso de  $v_i = "1"$ ,  $i_{C1}$  es mayor que  $i_{C2}$ , cae más tensión en  $R_{C1}$  que en  $R_{C2}$ , y estas caídas producen valores diferentes en  $v_{C1}$  y  $v_{C2}$  que se transmiten a  $T_3$  y  $T_4$ . La figura 4.4 muestra la característica de transferencia del inversor ECL.



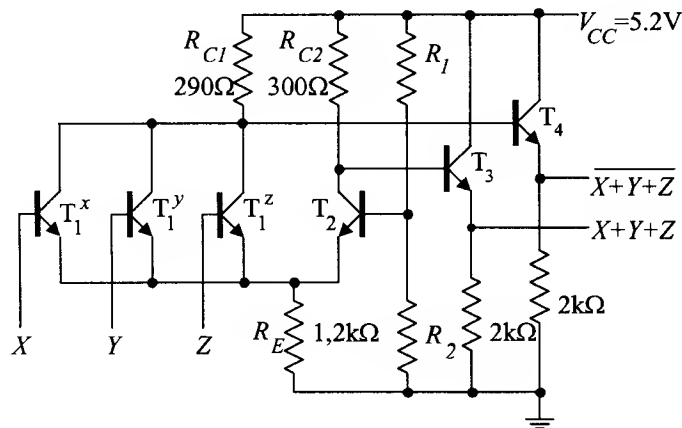
**Figura 4.4.** Características de transferencia de la salida inversora y no inversora del circuito inversor ECL

Obsérvese que el efecto de los transistores  $T_3$  y  $T_4$  es apoyar a tierra las salidas e introducir ganancias en corriente, pero no en tensión. Es decir, la tensión en el emisor de  $T_4$ ,  $\overline{v_o}(t)$ , es la misma que en el colector de  $T_1$ ,  $v_{C1}$ , menos la pequeña caída (del orden de  $0.6V$ ) en la unión base-emisor de

$T_4$  que está conduciendo. Lo mismo ocurre con la salida,  $v_o(t)$ , en el emisor de  $T_3$ , que reproduce la tensión de colector  $T_2$ ,  $v_{C2}$ , menos la caída en la unión base-emisor de  $T_3$ .

### 4.1.3 Puerta NAND en ECL

Si ya sabemos cómo funciona el circuito inversor, para obtener una puerta NOR nos basta con añadir más transistores en paralelo con  $T_1$ , tal como muestra el circuito de la figura 4.5.



**Figura 4.5.** Puerta OR-NOR en tecnología ECL

Los márgenes de ruido se encuentran en el rango de los  $150\text{ mV}$ . Sin embargo, dada la estructura de amplificador diferencial de ECL, con alto factor de rechazo al modo común, la inmunidad al ruido de transmisión es muy alta. Por eso se selecciona esta familia en aplicaciones de sistemas digitales de comunicación e instrumentación electrónica para transmisión de datos.

El retardo de propagación es el parámetro clave en ECL. Se consiguen valores del orden de  $1\text{ ns}$  con disipación de potencia de  $100\text{ mW}$ , dando lugar a soluciones específicas en todas aquellas aplicaciones de instrumentación, cálculo, comunicación o control en las que es imprescindible la velocidad. Tiene como inconvenientes el consumo relativamente alto y la necesidad de circuitos de adaptación para desplazar los niveles lógicos hasta hacerlos compatibles con los correspondientes a otras familias lógicas, tales como TTL ó MOS.

Las figuras 4.6, y 4.7 muestran los circuitos correspondientes a la adaptación de niveles lógicos e impedancias entre TTL y ECL cuando ambos se alimentan a  $+5\text{ V}$ . En la figura 4.6 se muestra una

forma sencilla de pasar de TTL a ECL mediante un partidor de tensión que atenúa las salidas TTL hasta el nivel ECL. En 4.7.a y b se muestran dos soluciones al problema inverso. La solución 4.7.a aprovecha la estructura de par diferencial de la familia ECL y usa sus salidas complementarias para alimentar a otro diferencial formado por dos transistores PNP. La solución 4.7.b usa un sólo transistor y es algo más lenta. El ejercicio E.4.3 del texto de problemas analiza con precisión estas etapas adaptadoras entre TTL y ECL y viceversa.

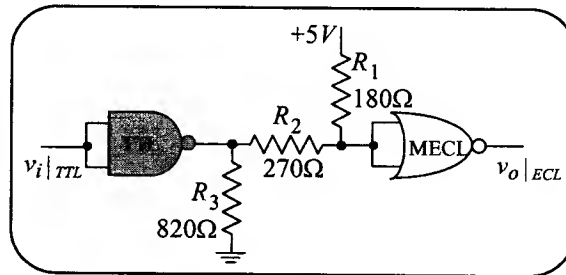


Figura 4.6. Circuitos para pasar de TTL a ECL.

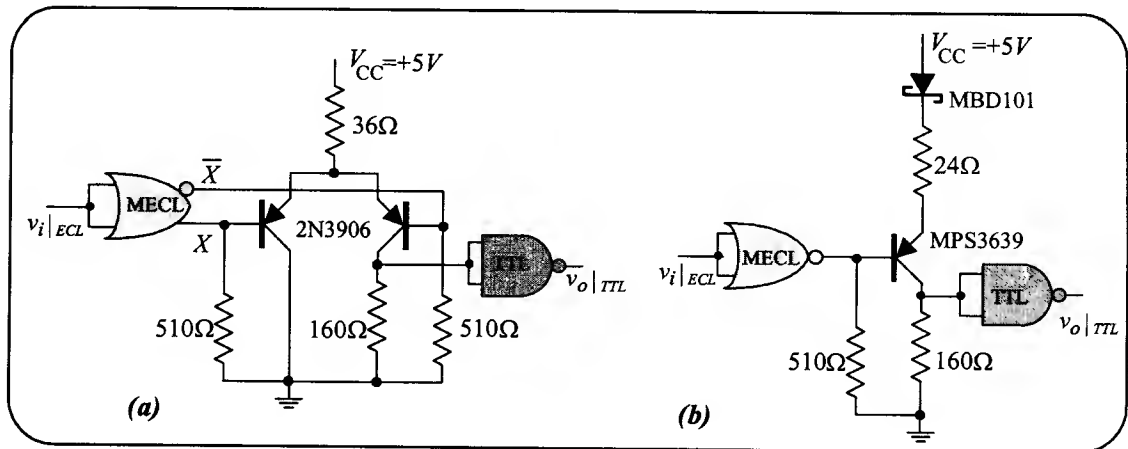


Figura 4.7. Adaptación de niveles entre ECL y TTL.

## 4.2. INVERSOR BÁSICO EN MOS

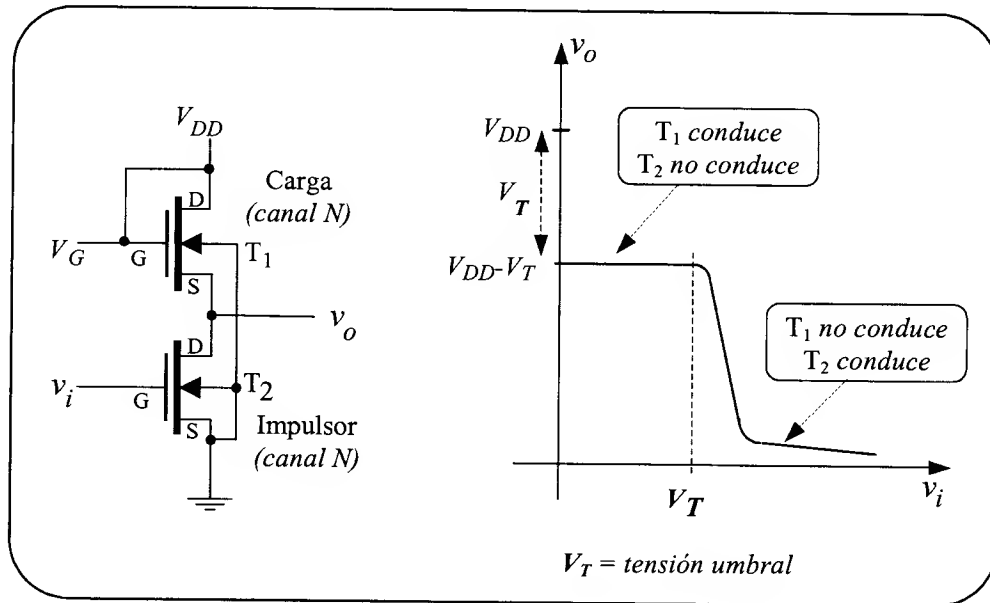
Todas las familias lógicas estudiadas hasta ahora, tanto las saturadas como las no saturadas, se

han realizado en tecnología bipolar. Vamos a estudiar ahora los circuitos digitales basados en el transistor MOS. Las ventajas que suponen el diseño con transistores de efecto campo MOS respecto a las familias de tecnología bipolar se refieren a la *densidad de integración* y al *consumo*. Al conseguir la misma función lógica en un área mucho menor, con un proceso de fabricación más sencillo y un consumo más bajo se facilita la integración de funciones globales más complejas. De hecho la integración en gran escala (LSI) y en muy gran escala (VLSI) está dominada por las tecnologías MOS y CMOS.

En este apartado y en el siguiente vamos a estudiar las puertas básicas en MOS y CMOS, su estructura interna, sus características de transferencia y las formas de conexión necesarios para conseguir puertas NAND y NOR y puertas de transmisión, en el caso del CMOS.

La tecnología N-MOS y P-MOS se usa poco en la síntesis de circuitos integrados en lógica combinacional, donde la familia dominante a partir de cierto grado de integración es la CMOS. No obstante su uso en lógica secuencial y en lógica dinámica es frecuente. Veremos más adelante, en los temas dedicados al estudio de las memorias RAM dinámicas y los registros CCD como estas estructuras MOS permiten realizar prácticamente todas las funciones necesarias en circuitos secuenciales (almacenamiento de carga en capacidades, inyección, extracción y desplazamiento de paquetes de carga, etc...).

Empecemos ya a estudiar el inversor MOS. Su estructura es análoga a la del inversor básico en tecnología bipolar (RTL) sólo que ahora el transistor impulsor ("driver") es MOS y la resistencia de carga,  $R_L$ , se realiza en tecnología integrada mediante otro transistor operando en zona triodo o zona de saturación y con el sustrato, en general, unido a tierra. Esto permite obtener fácilmente resistencias integradas del orden de los  $200\text{k}\Omega$  sobre un área mucho menor que las usadas en resistencias de difusión. La *figura 4.8* muestra el esquema para un inversor en el que los dos transistores son de canal N con sustrato común y que trabajan en modo de realce. En la misma figura se incluye también la característica de transferencia y el valor de la tensión de salida en función de la tensión umbral,  $V_T$ . Hasta que la tensión de puerta en el transistor impulsor,  $V_G$ , no alcanza el valor  $V_T$  el transistor no conduce y cuando lo alcanza conduce y se comporta para situaciones estacionarias como una resistencia. En condiciones dinámicas tendríamos que considerar una capacidad equivalente al conjunto de los efectos capacitivos en las uniones.



**Figura 4.8.** Inversor MOS. a) Circuito. b) Característica de transferencia.

Hay otras muchas posibilidades de diseñar inversores MOS combinando para  $T_1$  y  $T_2$  las posibles opciones de canal P o canal N y modo de realce o modo de vaciamiento tal como se muestra en la tabla de la *figura 4.9*.

TIPO	IMPULSOR			CARGA		
	CANAL	MODO	ZONA	CANAL	MODO	ZONA
PELT	P	realce	triado	P	realce	triado
PELS	P	realce	triado	P	realce	saturación
PDLS	P	realce	triado	P	vaciamiento	saturación
NELT	N	realce	triado	N	realce	triado
NELS	N	realce	triado	N	realce	saturación
NDLS	N	realce	triado	N	vaciamiento	saturación
CMOS	N	realce	triado	P	realce	triado

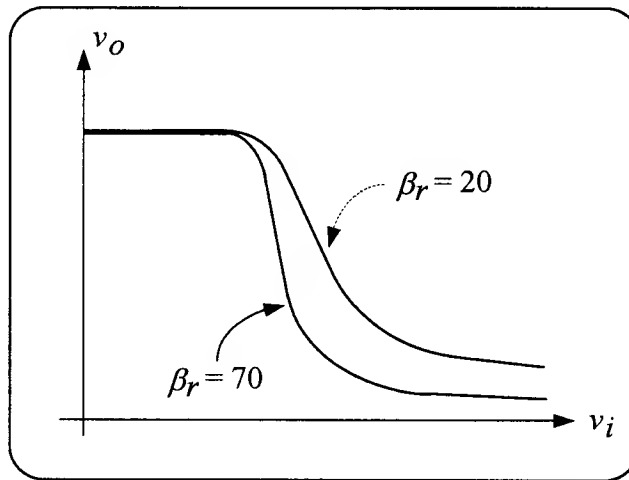
**Figura 4.9.** Posibles configuraciones de inversores MOS dependiendo del tipo y modo de los transistores impulsor y de carga (canal P ó N y modo de realce o modo de vaciamiento).

El transistor que actúa como impulsor siempre está en la región triodo cuando la entrada está en alta [ $v_i = v("1")$ ]. En cambio, el transistor que simula la carga puede estar en zona triodo o en zona de saturación. En todos los casos las características de transferencia de los inversores dependen críticamente de la geometría del canal, en especial del parámetro  $\beta_r$  (figura 4.10).

$$\beta_r = \frac{g_m(\text{impulsor})}{g_m(\text{carga})} = \frac{\beta V_D(\text{impulsor})}{\beta V_D(\text{carga})} = \frac{\frac{W}{L}(\text{impulsor})}{\frac{W}{L}(\text{carga})} \quad [4.12]$$

donde  $L$  es la longitud del canal y  $W$  el ancho de la zona de transición, tal como vimos en el tema 2 al estudiar los transistores MOS. Quizas sea conveniente ver la figura 2.33 donde se muestra la estructura de los transistores MOS de canal N.

Esto es válido para PELT, PELS y PDLT, inversores llamados de tipo proporcional, en los que al aumentar  $\beta_r$  aumenta el recorrido lógico [ $v("1") - v("0")$ ], para un mismo valor de la fuente de alimentación. Veremos más adelante que en CMOS la  $v_o$  no depende de  $\beta_r$  por ser un circuito simétrico en el que las funciones de transistor impulsor, ( $T_2$ ) y transistor carga ( $T_1$ ) son intercambiables.



**Figura 4.10.** Influencia de la geometría en las características de transferencia.

Antes de iniciar el análisis del inversor NMOS queremos recordar la nomenclatura que usamos para las tensiones. Cuando hablamos de  $V_D$ ,  $V_G$  y  $V_S$  nos referimos a la tensión del drenador, la puerta

y la fuente referente a tierra. Sin embargo, cuando usamos dos letras en los subíndices,  $V_{DS}$  y  $V_{GS}$ , nos referimos a las diferencias de tensión entre dos terminales concretos del transistor. Así,  $V_{DS}=V_D-V_S$ , es la diferencia de tensión entre los terminales de drenador y fuente. Análogamente,  $V_{GS} = V_G - V_S$ , es la diferencia de tensión entre la puerta y la fuente. En el transistor impulsor,  $T_2$ , del circuito inversor en MOS, la tensión  $V_{GS}$  coincide con  $V_G$  porque la fuente está a tierra ( $V_S = 0$ ), pero en el transistor de carga,  $T_1$ ,  $V_G$  es:

$$V_{GS}(de T_1) + V_S(de T_1) = V_{GS}(de T_1) + V_D(de T_2) \quad [4.13]$$

El circuito de la *figura 4.8* es un inversor NMOS tipo NELS porque los dos transistores son de canal  $N$  y, además, trabajan en modo de realce. El transistor  $T_2$  actúa como *impulsor* trabajando en región triodo y el transistor  $T_1$  se llama transistor de *carga* porque hace las funciones de la resistencia de carga,  $R_D$ . Este transistor actúa siempre en la región de saturación. Veamos por separado el comportamiento de estos dos transistores.

#### 4.2.1. Comportamiento del transistor de carga

Veamos primero cómo se comporta el transistor  $T_1$  que actúa como carga dinámica de  $T_2$ . La *figura 4.11* nos recuerda lo estudiado en el tema segundo representando las tres regiones de trabajo de un transistor NMOS. En condiciones generales, en las que  $V_{GS}$  varía, todo transistor NMOS en modo de realce empieza a conducir cuando su tensión de puerta supera un cierto valor umbral,  $V_T$ , ( $V_{GS} \geq V_T$ ) y lo hace sobre la curva característica correspondiente a ese valor de  $V_{GS}$ , pasando primero por las zonas lineal y cuadrática (región triodo) y entrando después en saturación, cuando  $V_{DS}$  vale,

$$V_{DS} = V_{DSsat} = V_{GS} - V_T \quad [4.14]$$

A partir de ese momento, la corriente permanece constante en su valor de saturación

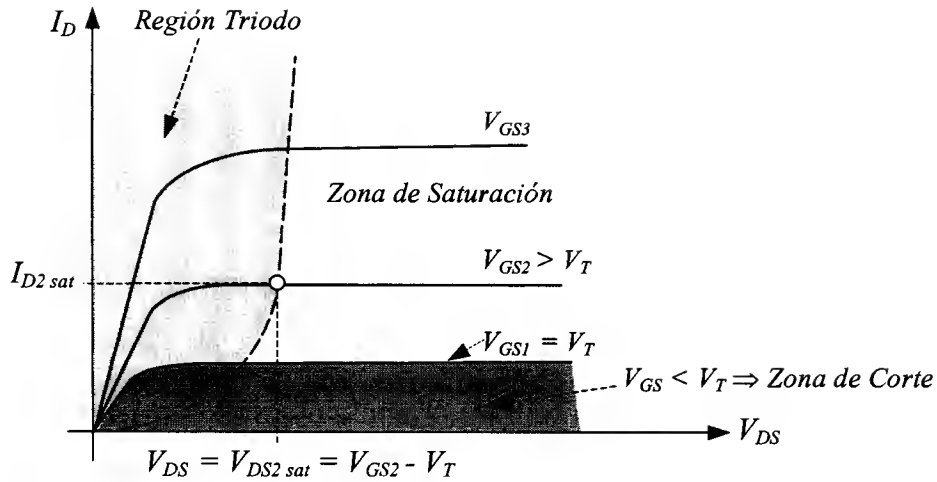
$$I_{Dsat} = K^* (V_{GS} - V_T)^2 = K^* (V_{DSsat})^2 \quad [4.15]$$

Así, para cada curva  $V_{GS} = cte.$ , con valores ( $V_{GS1}$ ,  $V_{GS2}$ , ...,  $V_{GSn}$ ), hay un punto en el que se cumple

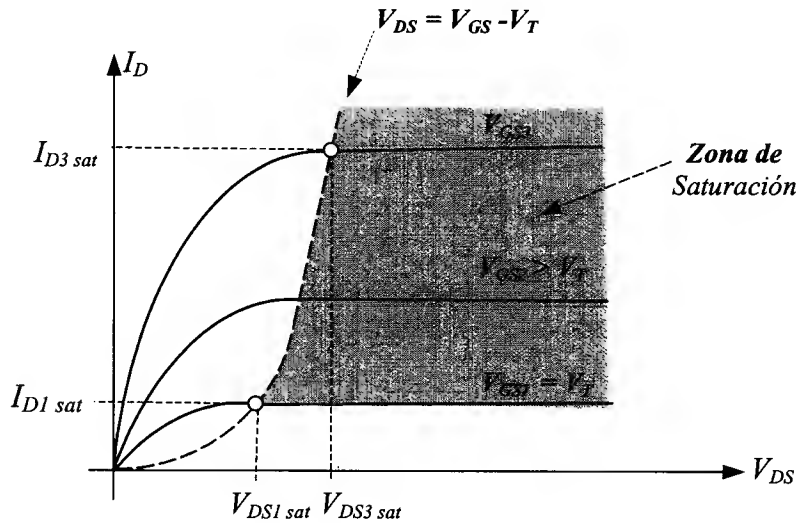
$$V_{DS} = V_{GS} - V_T \quad \text{é} \quad I_D = K^* (V_{DSsat})^2 \quad [4.16]$$

El lugar geométrico de todos estos puntos que definen la entrada en zona de saturación para los distintos valores de la tensión de puerta  $V_{GS}$  es el que se muestra en la curva de trazos de

la figura 4.12.



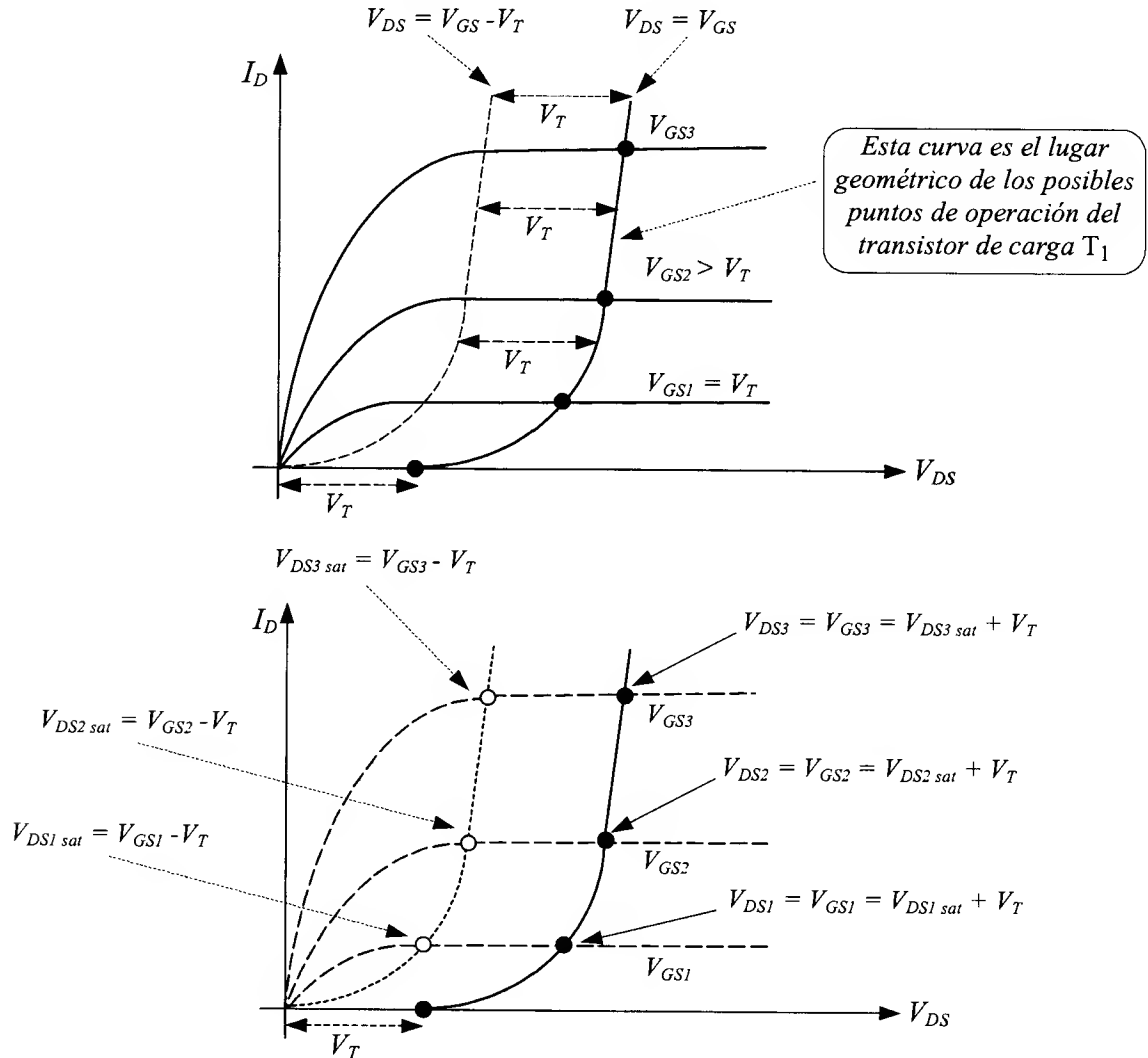
**Figura 4.11.** Regiones de trabajo de un transistor NMOS



**Figura 4.12.** Lugar geométrico de los puntos de entrada en zona de saturación para las distintas curvas características correspondientes a valores crecientes de  $V_{GS}$



Veamos ahora por qué esta curva de puntos de la *figura 4.12* está desplazada de la forma que se muestra en la *figura 4.13*, donde la curva original se ha dibujado en trazo fino intermitente y la curva desplazada, que representa el lugar geométrico de los posibles puntos de operación del transistor de carga T1, se ha dibujado en trazo grueso.

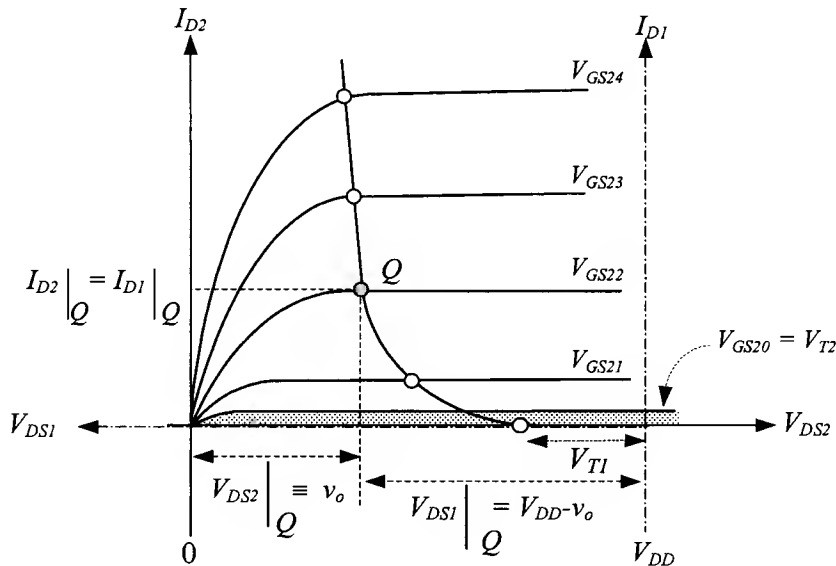


**Figura 4.13.** (a) Desplazamiento de la curva  $V_{DS} = V_{GS} - V_T$ . (b) Obsérvese que para cada valor de la corriente de drenador, sólo hay un punto de trabajo posible.

En condiciones normales las curvas características comienzan en el origen. Es decir, para valores muy pequeños de  $V_{DS}$  ya hay conducción [con  $I_D = K(V_{GS} - V_T) V_{DS}$ ], con tal que  $V_{GS} \geq V_T$ . ¿Qué pasa cuando unimos la puerta con el drenador para sintetizar una resistencia de carga mediante un transistor NMOS?. Pues que al hacer  $V_{DS} = V_{GS}$ , la condición de mínimo se transfiere ahora a la  $V_{DS}$ . Es decir, como el transistor no conduce hasta que  $V_{GS} \geq V_T$  y  $V_{GS} = V_{DS}$ , el transistor no conducirá hasta que  $V_{DS}$  no alcance el valor  $V_T$ . Por eso la curva de puntos ( $V_{DS} = V_{GS} - V_T$ ) se desplaza ahora  $V_T$  hacia la derecha y empieza en  $V_{DS} = V_T$ , en vez de empezar en el origen.

#### 4.2.2. Comportamiento del Transistor Impulsor

Hemos visto el comportamiento del *transistor de carga*,  $T_1$  y el lugar geométrico de sus posibles puntos de trabajo. Veamos ahora el funcionamiento del *transistor impulsor*  $T_2$  que está en serie con  $T_1$  y, por consiguiente, tienen el mismo valor de corriente de drenador ( $I_{D1} = I_{D2}$ ). La *figura 4.14* muestra la obtención por un procedimiento gráfico del punto de trabajo.



**Figura 4.14.** Punto de trabajo de  $T_2$ .

Este transistor  $T_2$  no tiene ninguna limitación porque su terminal de fuente no está unido al drenador. Por consiguiente, su punto de trabajo puede estar en cualquier posición sobre sus curvas

características,  $I_{D2} = f(V_{DS2}, V_{GS2})$ . El punto en el que finalmente trabajará es aquel que cumpla a la vez las condiciones de  $T_2$  y las de  $T_1$ . Es decir, estará trabajando en la *intersección* de la curva vista anteriormente ( $V_{GS} = V_{DS}$ ) con la característica de  $T_2$  definida por su tensión de puerta ( $V_{GS}$ ). Esto es lo que ilustra la *figura 4.14*. Lo que hemos hecho en esta figura es superponer las gráficas correspondientes a  $T_1$  con las correspondientes a  $T_2$  de forma que la curva sobre la que trabaja  $T_1$  es la imagen especular de la que obtuvimos antes y está apoyada a un eje que empieza en  $V_{DD}$  (tensión de alimentación del circuito) puesto que ambos transistores están conectados en serie.

Este procedimiento de obtención del punto de trabajo ( $Q$ ) es el mismo que vimos al estudiar la polarización de los transistores bipolares con una resistencia en el colector ( $R_C$ ), de forma que por un lado debía cumplirse la ley de Ohm, ( $E = V_{CE} + I_C R_C$ ) y por otro lado debía cumplirse la relación [ $I_C = f(V_{CE}, I_B)$ ]. Por consiguiente, el transistor sólo podía estar en aquel punto ( $I_{CQ}, V_{CEQ}$ ) que cumpliera a la vez ambas condiciones. La solución gráfica a este sistema de dos ecuaciones era la intersección de la recta de carga con la curva característica.

Aquí hemos hecho lo mismo sólo que sustituyendo  $R_C$  por el transistor  $T_1$  que emula una resistencia de valor,

$$R_{DS1} = \frac{V_{DS}}{I_{DS}} = \frac{V_{DS}}{K^* V_{DS}^2} = \frac{1}{K^* V_{DS}} \quad [4.17]$$

Ahora, la tensión de alimentación se reparte entre ambos transistores:

$$V_{DD} = V_{DS1} + V_{DS2} = R_{DS1} I_{DS2} + V_{DS2} \quad [4.18]$$

y por otro lado debe cumplirse la condición

$$V_{DS2} = f(I_{DS2}, V_{GD2}) \quad [4.19]$$

por lo que la solución gráfica de este sistema de ecuaciones nos da el punto de trabajo ( $Q$ ) del inversor.

### 4.2.3. Curva de Transferencia del Inversor

Veamos ahora la curva de transferencia del inversor. Para ello partimos de una señal de entrada nula,  $v_i = 0$ , y la vamos aumentando hasta el valor máximo  $V_{DD}$  observando en cada caso el valor de la salida. Así, para  $v_i \approx 0$ ,  $T_2$  estará al corte porque su tensión de puerta no supera la tensión umbral  $V_{T2}$ ,

presentando por tanto alta impedancia y no dejando pasar corriente

( $I_{D1} = I_{D2} \approx 0$ ) y, por consiguiente, si no cae nada en  $R_{DS1}$ ,

$$v_o \equiv V_{DS2} = V_{DD} - I_{D2} R_{DS1} \approx V_{DD} \quad [4.20]$$

Cuando la tensión de entrada,  $v_i$ , supera el valor umbral,  $V_{T2}$ , el transistor  $T_2$  empieza a conducir en su zona lineal con corriente creciente a medida que aumenta  $v_i$  y el transistor de carga está en saturación en el punto correspondiente a cada uno de esos valores de corriente ya que  $I_{D1} = I_{D2} = I_D$ .

Al seguir aumentando  $v_i$ ,  $T_2$  conduce en región triodo con mayor valor de  $I_D$ , y con mayor caída en  $T_1$ , con lo que disminuye la tensión de salida  $v_o$ .

$$v_o = V_{DS2} = V_{DD} - I_{DS2} R_{DS1} \quad [4.21]$$

$$\text{Si } v_i \uparrow, I_{DS2} \uparrow, I_{DS1} \uparrow, (I_{DS2} R_{DS1}) \uparrow \Rightarrow v_o \downarrow \quad [4.22]$$

Durante la travesía de esta zona de transición  $T_1$  sigue saturado, sólo que cada vez con un valor mayor de su corriente de drenador.

Así siguen las cosas hasta que la tensión de entrada  $v_i$  se acerca al valor de la tensión de alimentación  $V_{DD}$ . En el momento en que  $V_{DS1}$  no alcance el valor umbral  $V_{T1}$ , el transistor  $T_1$  pasará al corte ofreciendo una impedancia muy alta en la que cae toda la tensión  $V_{DD}$ , de forma que la tensión de salida es prácticamente cero. Este bajo valor de la tensión de salida coincide con la conducción de  $T_2$ , en su valor más alto de corriente.

$v_i \geq V_{DD} - V_{T1}$  ▶  $T_1$  se corta y  $R_{DS1} \uparrow$

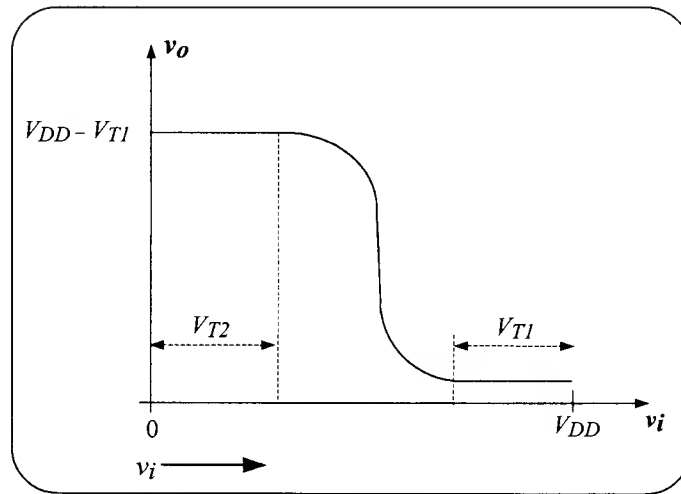
▶  $T_2$  conduce con  $I_{DS2} \uparrow$

Así,

$$v_o \equiv V_{DS2} = V_{DD} - I_{DS2} R_{DS1} \approx 0$$

$$[4.23]$$

Así la curva de transferencia es la que se muestra en la *figura 4.15*.



**Figura 4.15.** Característica de transferencia del inversor NMOS.

#### 4.2.4. Puertas NAND y NOR con NMOS

La *figura 4.16* muestra las conexiones necesarias para obtener las funciones NAND y NOR con MOS, junto con las tablas de verdad correspondientes.

La operación de ambos circuitos se comprende fácilmente recordando que los transistores de carga actúan como resistencias. Es decir, las puertas NAND y NOR en MOS operan como las correspondientes en RTL, cambiando  $R_L$  por un transistor. En el caso NOR, al estar los dos impulsores en paralelo bastará con que uno tenga la entrada en alta para que entre en conducción y la tensión en su drenador sea la correspondiente a baja.

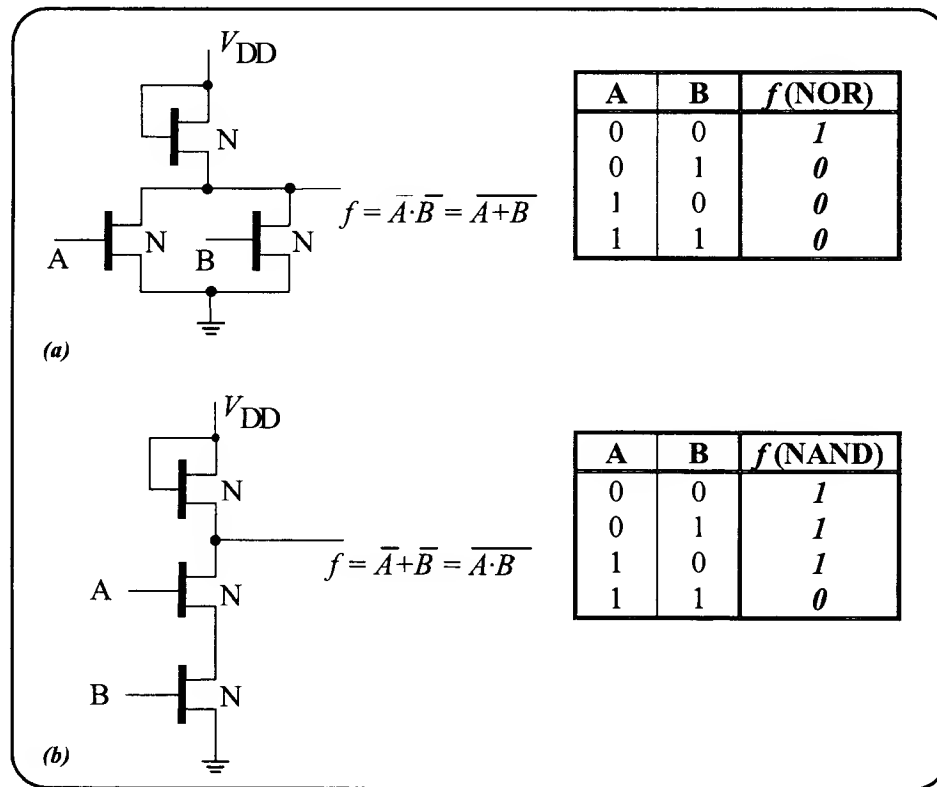
Por construcción, todos los impulsores tienen la misma  $W/L$  y por consiguiente la misma conductancia  $g_m$ . La  $g_m$  de una cualquiera de las entradas es suficiente para que  $I_D/g_m$  proporcione la tensión de drenador deseada en conducción [ $v_o = V_D(ON) = v("0")$ ]. La acción de las otras ramas paralelo no cambia  $g_m$ , sólo tiende a bajar  $V_D(ON)$ .

Un razonamiento análogo sirve para la puerta NAND que de nuevo repite la estructura RTL sustituyendo  $R_L$  por un transistor. Tal como se ilustra en la *figura 4.16.b*, la puerta NAND consta de tres transistores MOS en serie, dos haciendo función de impulsores y el tercero actuando como carga activa. Al estar los impulsores en serie, sólo cuando las entradas en las puertas *A* y *B* están en alta, hay

conducción entre  $V_{DD}$  y tierra y, por consiguiente, la salida  $f$  está en baja tal como corresponde a una puerta NAND.

Inversamente, cuando los impulsores  $A$  y  $B$  están en paralelo (parte (a) de la figura 4.16) basta con que uno de ellos esté en conducción para que la salida esté en baja. Sólo cuando las dos entradas están en baja los transistores impulsores no conducen, no hay corriente en el transistor de carga y la tensión de salida coincide, prácticamente, con la tensión de alimentación,  $V_{DD}$ , tal como corresponde a una puerta NOR.

Para aumentar el número de entradas a una puerta NAND o una puerta NOR basta con incluir más transistores impulsores en serie o en paralelo, manteniendo la estructura de la figura 4.16.



**Figura 4.16.** Puertas NAND y NOR en MOS. (a) NOR. (b) NAND.

### 4.3. INVERSOR BÁSICO EN CMOS

Las características de una *familia lógica ideal* son:

1. Que no consuma potencia.
2. Retardo de propagación nulo (velocidad infinita).
3. Tiempo de subida (rise time) y bajada (fall time) controlables.
4. Inmunidad al ruido del 50% del rango lógico:  $\frac{V_{alta} - V_{baja}}{2}$

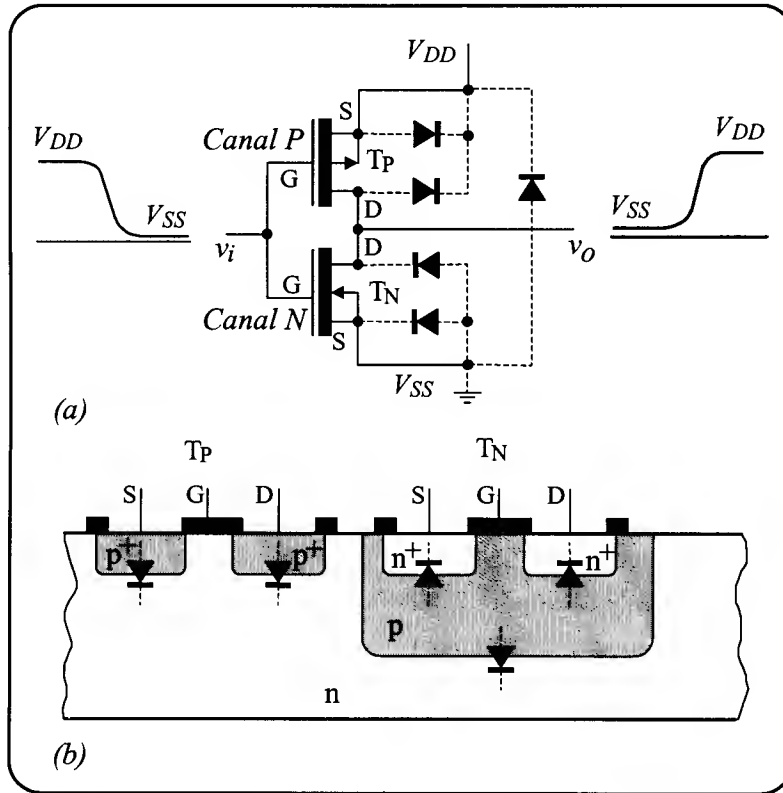
En general no se ha desarrollado ninguna familia lógica que proporcione una solución óptima. Lo distintivo de la familia CMOS, (MOS con transistores complementarios), es que se aproxima al ideal globalmente. CMOS es la solución de compromiso que ofrece las siguientes características:

- Necesita un *consumo mínimo* para mantener cualquiera de los dos estados estables (menor de 10 nW), ya que siempre hay un dispositivo que no conduce. Sólo hay consumo en las transiciones y este consumo depende de la alimentación y de la frecuencia.
- Los *retardos de propagación son pequeños* (inferiores a 25ns) y, junto con los tiempos de subida y bajada, dependen de  $V_{DD}$ .
- Hay *alta inmunidad al ruido*, pues vale justo la mitad del rango dinámico y éste coincide con el valor de la tensión de alimentación.
- Hay *flexibilidad* en los valores de  $V_{DD}$  y  $V_{SS}$ , ( $5V < V_{DD} < 18V$ ).
- Permite *gran conectividad* a la entrada ("fan-in") y a la salida ("fan-out"), debido a las altas impedancias de entrada inherentes al circuito. La puerta de un transistor MOS es prácticamente un aislante por el que no pasa corriente continua y, además, la entrada a un inversor CMOS es a través de una puerta común a dos transistores.
- *Facilidad de fabricación y pocas exigencias de área* lo que facilita la integración a gran escala.

#### 4.3.1. Comportamiento Estático

El proceso de fabricación se basa en la formación de un par de transistores MOS, canales P y N,

modo de realce y con difusiones complementarias para aislar los canales, tal como se ilustra en la *figura 4.17*. En línea a trazos incluimos las uniones P-N de carácter parásito. Estos diodos parásitos están polarizados en sentido inverso y son los únicos responsables del consumo en los estados estables.



**Figura 4.17.** Estructura del inversor CMOS. (a) Circuito. (b) Corte transversal de la estructura física que lo sintetiza.

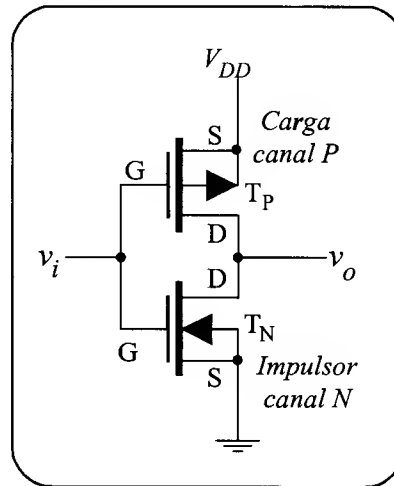
Este inversor combina la rápida conmutación asociada a un MOS en conducción (saturación o triodo) con un bajo consumo, debido a que siempre existe uno de los transistores (canal P o N) que no conduce. Los niveles lógicos son, prácticamente,  $V_{DD} = "1"$  y  $V_{SS} = "0" = 0 \text{ V}$ .

Analicemos el esquema básico sin considerar los elementos parásitos. Por comodidad, repetimos en la *figura 4.18* el circuito simplificado, sin diodos parásitos y resaltando el tipo de canal



del impulsor y de la carga. En el análisis vamos a seguir los siguientes pasos:

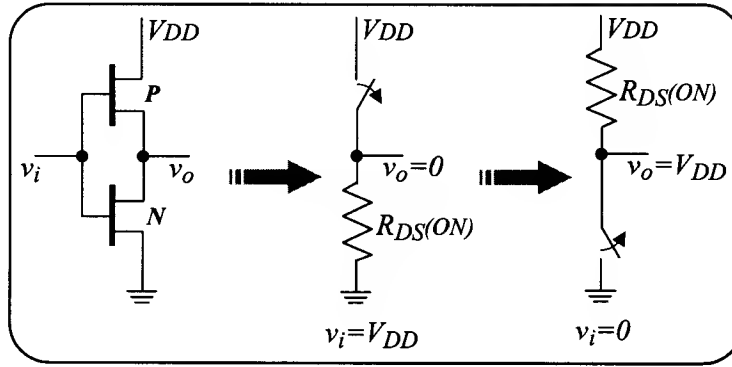
1. Calcular la característica de transferencia de  $T_N$ .
2. Calcular la característica de transferencia de  $T_P$ .
3. Superponerlas en las dos situaciones extremas: ( $T_P$  en corte,  $T_N$  en conducción) y ( $T_P$  en conducción,  $T_N$  en corte).
4. Recorrer con  $v_i(t)$  el camino desde 0 hasta  $V_{DD}$ .



**Figura 4.18.** Versión simplificada del inversor CMOS.

Antes de empezar el análisis podemos obtener una idea intuitiva y rápida sobre cómo funciona el inversor CMOS usando su modelo como conmutador. Un transistor MOS (en continua) se comporta como un conmutador que para tensiones de puerta tales que  $|V_{GS}| < |V_T|$  no conduce (es un circuito abierto ideal) y cuando la tensión de puerta supera un valor umbral ( $|V_{GS}| > |V_T|$ , por ejemplo) el MOS conduce comportándose como una resistencia de valor  $R_{DS}$  (ON). Así, cuando  $v_i$  es alta (aproximadamente  $V_{DD}$ ), el transistor NMOS conduce y el PMOS está en circuito abierto, con lo que la salida  $v_o$  está en baja ( $v_o=0$ ). Inversamente, cuando la entrada  $v_i$  está en baja el transistor NMOS está en corte y el PMOS conduce, quedando entonces la salida apoyada a la tensión de alimentación,

$V_{DD}$ , tal como se muestra en la *figura 4.19*. Además, al no cerrar circuito en ninguno de los dos casos, no hay consumo en situaciones estacionarias.



**Figura 4.19.** Ilustración del funcionamiento del inversor CMOS usando para los transistores de canal P y N el modelo elemental de un conmutador que pasa de corte a conducción con una resistencia  $R_{DS}(ON)$  cuando la tensión de puerta supera el valor umbral  $V_T$ .

Veamos ahora un análisis más detallado recordando los conceptos de polarización, recta de carga y punto de trabajo de un transistor. En transistores bipolares y MOS con carga resistiva teníamos *rectas de carga* que se superponían a las curvas características. En inversores tipo NMOS, como el visto en el apartado anterior, sustituíamos la resistencia de carga por un transistor con la puerta unida al drenador, con lo que obteníamos una “*curva*” de carga que se superponía a las características del transistor impulsor. Ahora, en el inversor CMOS, tenemos dos familias de curvas características superpuestas y una única corriente de drenador que fija el punto  $Q$ .

Como puede observarse en la *figura 4.18* el inversor CMOS es análogo al NMOS sólo que ahora el transistor de carga es de *canal P*, mientras que el impulsor sigue siendo de *canal N*. Además, ahora los drenadores y las puertas de los transistores están unidos entre sí, de forma que ambos transistores pueden trabajar en cualquiera de las distintas zonas de funcionamiento, verificándose que:

$$V_{GSN} = v_i \quad \text{y} \quad V_{DSN} = v_o, \quad [4.24]$$

$$V_{GSP} = V_{DD} - v_i \quad \text{y} \quad V_{DSP} = V_{DD} - v_o. \quad [4.25]$$

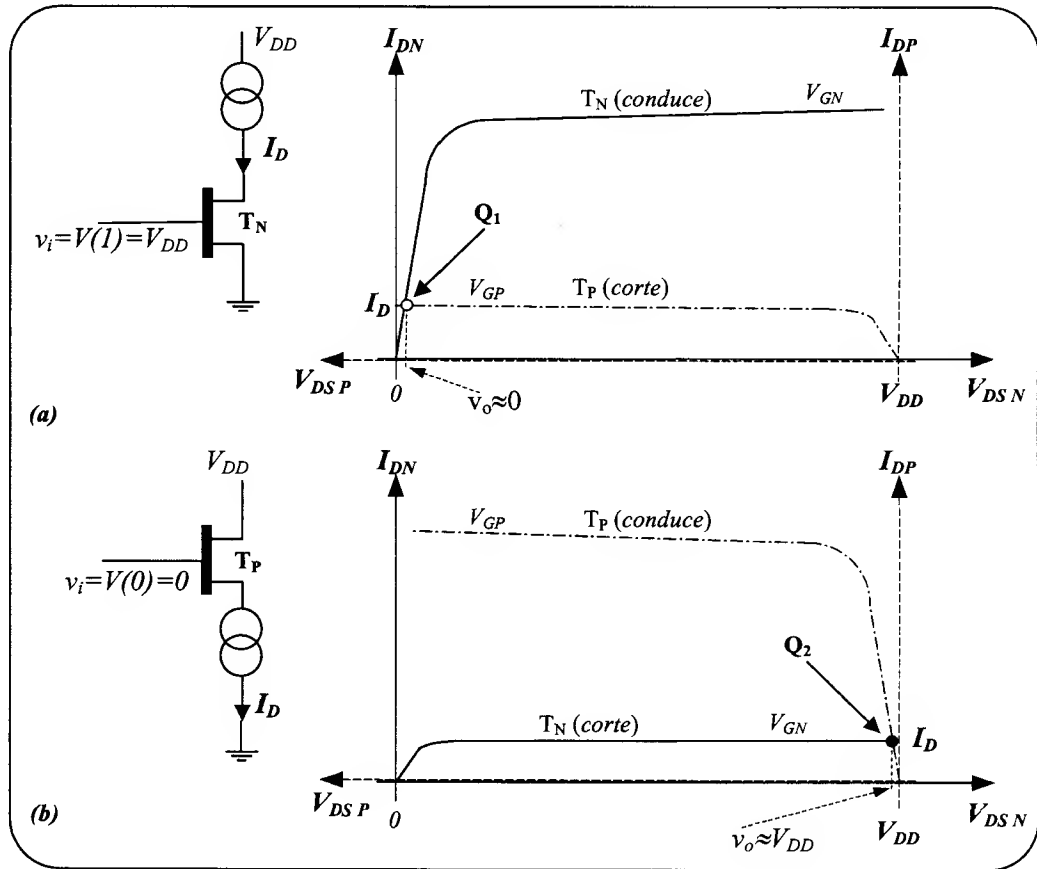
Como los dos transistores están en serie se debe verificar que:

$$V_{DSP} + V_{DSN} = V_{DD} \quad [4.26]$$

$$I_{DP} = I_{DN} = I_D.$$

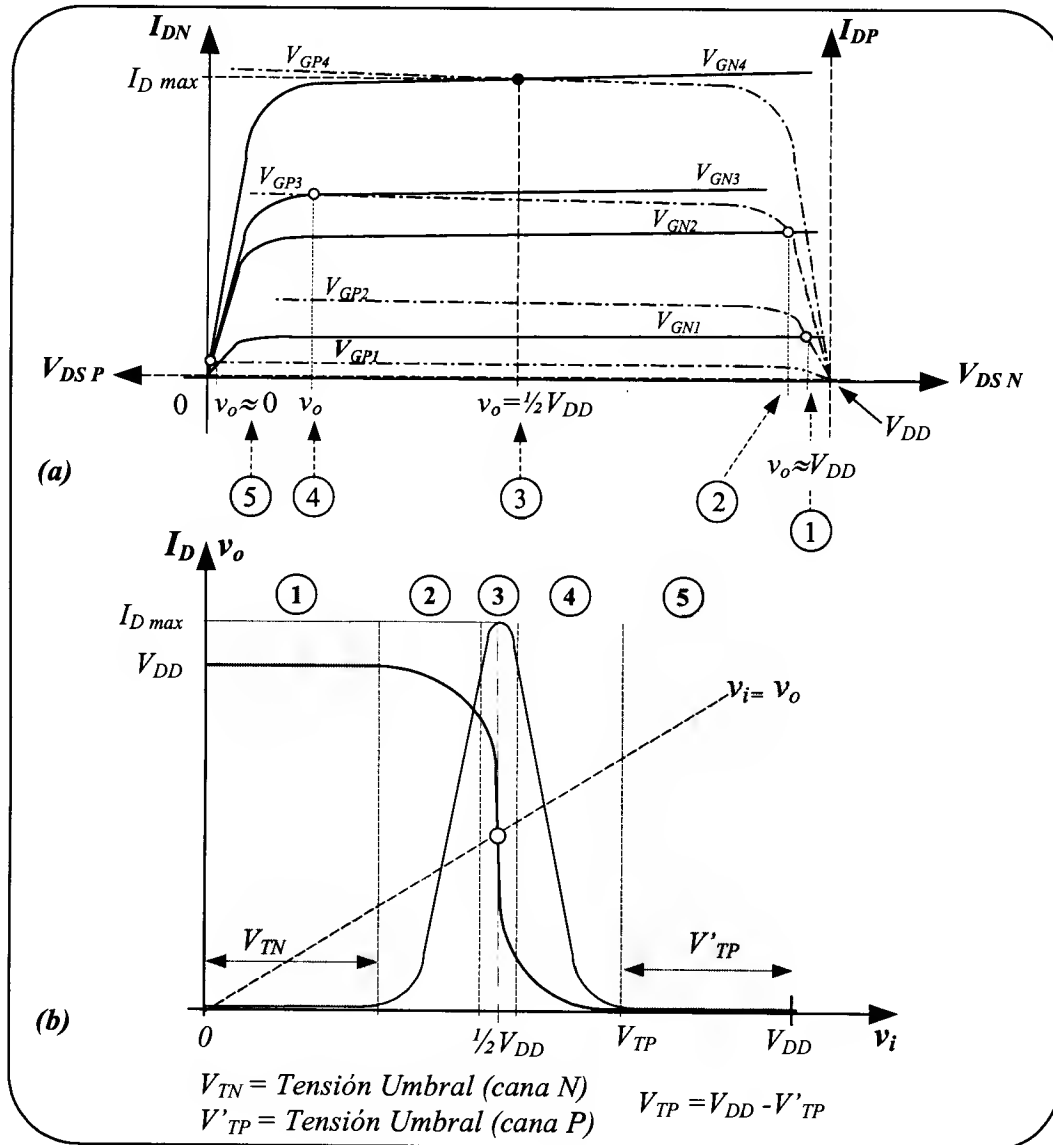
[4.27]

Los puntos de trabajo de ambos transistores vienen determinados por las intersecciones de sus curvas características ya que las del transistor de carga hacen la función de la recta de carga que hemos visto anteriormente. Así, la forma de resolver el problema gráficamente es superponer las dos familias de curvas características, pero las del transistor de canal P desplazadas  $V_{DD}$  (de la misma forma que hicimos con la curva de carga en el caso del inversor NMOS) como se muestra en la figura 4.20.



**Figura 4.20.** Curvas características de los transistores  $T_N$  y  $T_P$ . Para obtener los puntos  $Q$ , se desplazan las curvas características  $V_{DD}$  y se marcan las intersecciones. Así, el punto  $Q_1$  de  $T_N$  es  $(I_D, 0)$  y el de  $T_P$  es  $(I_D, V_{DD})$  (parte a de la figura). Obviamente la corriente es la misma porque  $T_P$  y  $T_N$  están en serie. La tensión,  $V_{DD}$ , se reparte entre los dos transistores.

Cuando  $v_o$  varía desde 0 hasta  $V_{DD}$ , los puntos de cruce de ambas características nos dan los posibles puntos de funcionamiento de los dos transistores. La *figura 4.21.a* muestra esta yuxtaposición de curvas y las cinco zonas de trabajo distinguibles entre 0 y  $V_{DD}$ . La *figura 4.21.b* muestra la característica de transferencia donde hemos identificado esas cinco zonas.



**Figura 4.21.** (a) Yuxtaposición de las curvas características de  $T_P$  y  $T_N$  para obtener los posibles puntos de trabajo. (b) Característica de transferencia del inversor CMOS.

En la característica de transferencia del circuito que se muestra en la *figura 4.21.b* hemos superpuesto la representación de la variación de  $I_D$  con  $v_i$ . Como podemos observar, el circuito realiza la función de inversión, presentando el mayor nivel de conducción para  $v_i = \frac{1}{2}V_{DD}$  que es cuando los dos transistores están saturados y presentan menor impedancia dejando pasar más corriente. Afortunadamente esta situación de consumo sólo ocurre en el proceso transitorio de conmutación. Podemos profundizar en la comprensión del inversor CMOS usando la tabla de la *figura 4.22* donde se describen los cinco estados funcionales por los que pasan  $T_P$  y  $T_N$  al recorrer el rango dinámico desde 0 hasta  $V_{DD}$ .

Zona	Transistor $T_N$	Transistor $T_P$
1	$v_i \equiv V_{GS} _N < V_{TN} \Rightarrow T_N \text{ cortado}$ $v_o \equiv V_{DS} _N \approx V_{DD}$ $I_D \approx 0$	$V_{GS} _P < V_{TP} \Rightarrow T_P \text{ conduce (zona óhmica)}$ $V_{DS} _P \approx 0$
2	$v_i \equiv V_{GS} _N \geq V_{TN} \Rightarrow T_N \text{ conduce (saturación)}$ $v_o \equiv V_{DS} _N \Rightarrow \text{disminuye}$ $I_D \Rightarrow \text{aumenta}$	$T_P \text{ sigue en conducción (región triodo)}$ $V_{DS} _P \Rightarrow \text{aumenta}$ $V_{TP} \equiv V_{DS} - V'_{TP} \quad V'_{TP} \equiv \text{Tensión Umbral canal P}$
3	$v_i \equiv V_{GS} _N > V_{TN} \Rightarrow T_N \text{ conduce (saturación)}$ $v_o \equiv V_{DS} _N \approx V_{DS} _P = \frac{1}{2}V_{DD}$ $I_D \Rightarrow \text{alcanza el valor máximo}$	$V_{GS} _P < V_{TP} \Rightarrow T_P \text{ conduce (saturación)}$ $V_{DS} _P \approx V_{DS} _N = \frac{1}{2}V_{DD}$
4	$\text{Sigue } \uparrow v_i \Rightarrow T_N \text{ conduce (región triodo)}$ $v_o \equiv V_{DS} _N \Rightarrow \text{disminuye}$ $I_D \Rightarrow \text{disminuye}$	$V_{GS} _P \leq V_{TP} \Rightarrow T_P \text{ conduce (saturación)}$ $V_{DS} _P \text{ aumenta}$
5	$v_i \equiv V_{DD} = V_{GS} _N \Rightarrow T_N \text{ conduce (reg. triodo)}$ $v_o \equiv V_{DS} _N \approx 0$ $I_D \approx 0$	$V_{GS} _P > V_{TP} \Rightarrow T_P \text{ cortado}$ $V_{DS} _P \approx V_{DD}$

**Figura 4.22.** Resumen de los distintos estados funcionales por los que pasan  $T_N$  y  $T_P$  a medida que la tensión de entrada,  $v_i$ , pasa desde 0 hasta  $V_{DD}$ . Comparar esta tabla con la *figura 4.21*.

### 4.3.2. Comportamiento Dinámico

Hemos visto el comportamiento estático del inversor, suponiendo que la entrada se encontraba de forma estacionaria en alta ( $V_{DD}$ ) o en baja ( $0V$ ). Veamos ahora qué pasa en los transitorios de conmutación que son los que limitan la velocidad máxima a la que puede trabajar un sistema digital basado en esta tecnología.

Cada proceso de conmutación real lleva asociado un retardo que es el tiempo que transcurre desde que cambia el valor de la entrada,  $v_i(t)$ , hasta que el cambio en la salida,  $v_o(t)$ , se estabiliza en un entorno prefijado de su valor estacionario final. Este retardo se debe básicamente a dos factores:

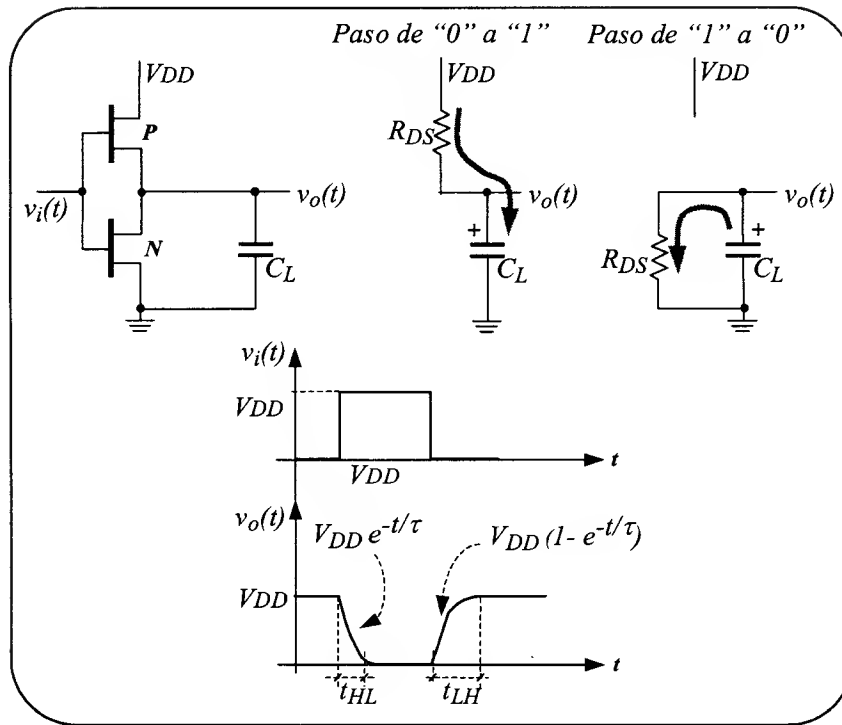
1. La existencia de un conjunto de capacidades de difusión e interelectródicas en los transistores PMOS y NMOS que no hemos considerado en el modelo resistivo anterior. En una primera aproximación estas capacidades se pueden representar mediante una única capacidad de carga,  $C_L$ , entre la salida,  $v_o(t)$ , y tierra.
2. El valor limitado de las corrientes de drenador y fuente de los transistores necesarias para cargar y descargar esas capacidades. Es decir, en el valor no despreciable de las resistencias  $R_{DS}(ON)$ .

Por consiguiente, el parámetro a minimizar es la constante de tiempo de carga y descarga de  $C_L$ ,  $\tau = R_{DS}(ON) C_L$ . Cuanto menor sea esa capacidad equivalente,  $C_L$ , y la resistencia de conducción de los transistores,  $R_{DS}(ON)$ , menor será el retardo asociado al transitorio de conmutación y mayor la velocidad de los circuitos CMOS correspondientes. La *figura 4.23* ilustra el proceso de carga y descarga de  $C_L$ .

Los tiempos de retardo en las transiciones de alta a baja ( $t_{HL}$ ) y de baja a alta ( $t_{LH}$ ) de la tensión de salida determinan la frecuencia máxima a la que puede funcionar el inversor. Su suma ( $t_{HL} + t_{LH}$ ) es el tiempo mínimo para completar un ciclo (pasar de “1” a “0” y, de nuevo, de “0” a “1”) y su inverso, la frecuencia máxima de conmutación

$$f_{max} = \frac{1}{t_{HL} + t_{LH}} \quad [4.28]$$

Para valores superiores de la frecuencia del reloj de un circuito digital síncrono, las puertas CMOS no pueden reaccionar adecuadamente a los cambios en sus tensiones de entrada y, por consiguiente, se comportan de forma errónea.



**Figura 4.23.** Transitorios de conmutación del inversor CMOS con los procesos de carga y descarga de la capacidad equivalente  $C_L$ .

### 4.3.3. Inversor de Tres Estados

Al igual que hicimos al estudiar la puerta NAND en TTL, también vamos a comentar cómo se pueden convertir los inversores CMOS en puertas de tres estados para facilitar la conexión cableada a un bus de  $n$  puertas y permitir la selección de una de ellas en cada momento, quedando las otras conectadas físicamente pero aisladas funcionalmente a través de un tercer estado de alta impedancia respecto a tierra y respecto a la fuente de alimentación,  $V_{DD}$ .

La idea es sencilla y se ilustra en el circuito de la *figura 4.24*. A un inversor CMOS convencional le añadimos dos nuevos transistores, uno de canal P en serie con la carga y otro de canal N en serie con el impulsor. Además, incluimos una entrada adicional para el control de la inhibición,  $C$ , de forma tal que cuando  $C$  está en baja los dos nuevos transistores no conducen y la salida  $v_o$  queda aislada. En cambio, cuando  $C$  está en alta, los dos nuevos transistores conducen con muy baja

impedancia, en serie con los originales, dejando paso así al modo usual de funcionamiento del inversor. Es decir ahora, con  $C=I$ ,  $v_o$  es “1” si  $v_i$  es “0” y viceversa.

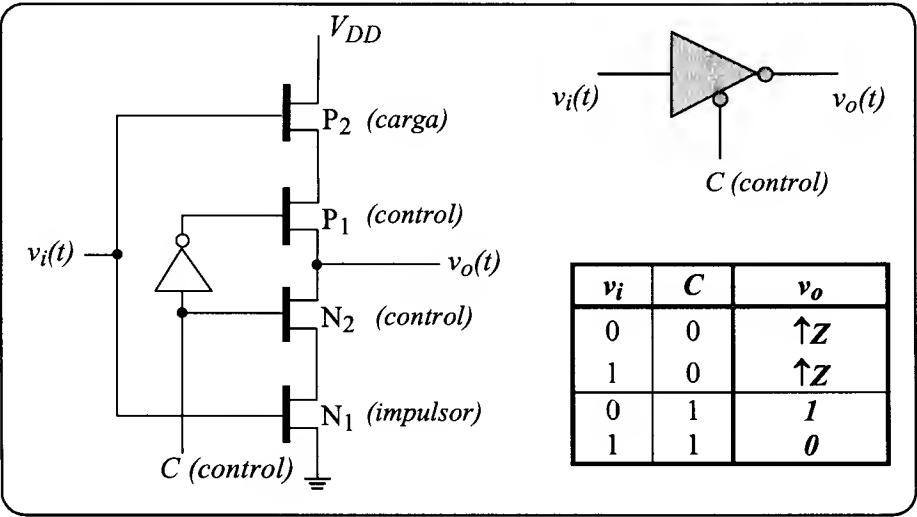


Figura 4.24. Modificación del inversor CMOS para incluir un tercer estado de alta impedancia.

### 4.3.4. Acoplo con Otras Familias

Generalmente cuando se diseñan sistemas digitales se suele usar una única familia lógica, tal como TTL, ECL, MOS o CMOS. Sin embargo, hay veces en que esto no es posible bien porque se quiera optimizar más de un parámetro o bien porque las subfunciones necesarias no existan en la familia lógica elegida, por lo que hay que recurrir a usar más de una familia lógica.

En estos casos nos encontramos con ciertos problemas de acoplo entre circuitos de familias diferentes puesto que cada uno funciona con valores distintos de tensiones, de corrientes y de impedancias. Nace así la necesidad de usar circuitos de acoplo o de interface para adaptar los distintos niveles de tensión y corriente. Ya hemos visto los circuitos de acoplo entre TTL y ECL. Veamos ahora el acoplo entre TTL y NMOS ó CMOS. Este tipo de acoplo es el más sencillo de todos, sobre todo cuando se usa 5V para la alimentación de los circuitos MOS. Así, para el caso de circuitos tipo NMOS se pueden conectar directamente sin necesidad de ningún circuito adicional.

En general, los parámetros que hay que tener en cuenta cuando se conectan puertas de distintas



familias lógicas son los valores de las tensiones umbrales para “0” y para “1”. Es decir, cada familia tiene un valor de la tensión de entrada por encima del cual interpreta que dicha señal de entrada es un “1” y otro valor por debajo del cual interpreta que la señal de entrada es un “0”. Estos valores se nombran con  $V_{iH}$  y  $V_{iL}$ . Análogamente ocurre con las corrientes de entrada para las que también existen dos valores umbrales,  $I_{iH}$  y  $I_{iL}$ . Cuando estas condiciones no se cumplen en las fronteras entre circuitos de distintas familias nos encontramos con problemas de funcionamiento, ya que un circuito puede interpretar que la señal de entrada es un “1”, cuando en realidad es un “0”, y a la inversa.

Lo mismo ocurre con la señal de salida. Existen unos valores umbrales para la tensión y la corriente de salida que marcan el funcionamiento correcto, de forma que, siempre que la tensión de salida supere el valor correspondiente a  $V_{oH}$ , podemos asegurar que el circuito responde con un “1” y siempre que esté por debajo del valor de  $V_{oL}$  se considera que responde con un “0”.

Así, podemos definir los siguientes parámetros:

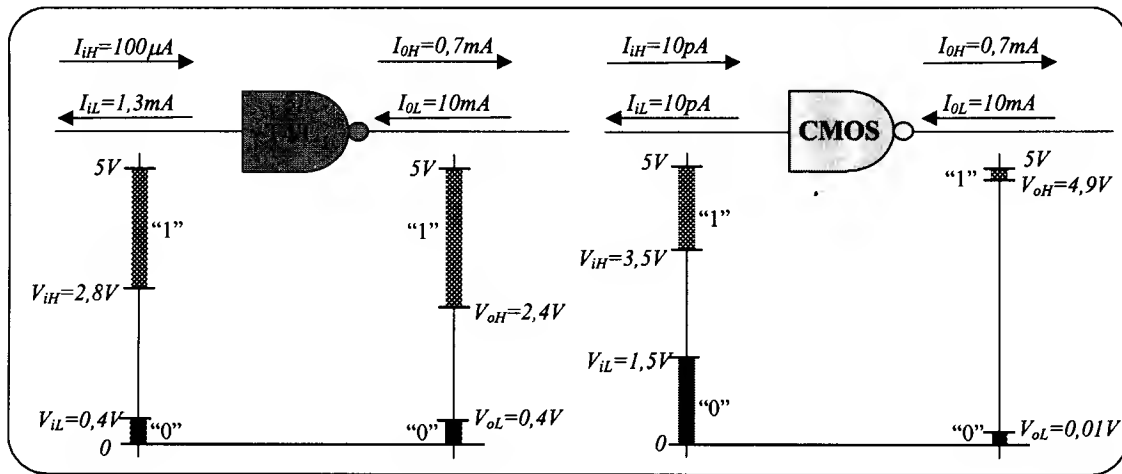
$V_{iH}$  = mínima tensión de entrada que la puerta reconocerá como un “1” sin ninguna ambigüedad  
 $V_{iL}$  = máxima tensión de entrada que la puerta reconocerá como un “0” sin ninguna ambigüedad  
 $V_{oH}$  = mínima tensión de salida de la puerta cuando se supone que responde con un “1”.  
 $V_{oL}$  = máxima tensión de salida de la puerta cuando se supone que responde con un “0”.

De igual forma, también existen valores umbrales de las corrientes de entrada y salida que se representan por  $I_{iH}$ ,  $I_{iL}$ ,  $I_{oH}$ ,  $I_{oL}$ . Estos parámetros son útiles cuando es necesario calcular el “fan-out” y “fan-in”, valores típicos de estos parámetros para TTL y CMOS son los que se muestran en la figura 4.25.

Como podemos observar, los valores de las tensiones de entrada y de salida para valores de “1” y de “0” no coinciden. Por tanto, vamos a ver qué pasa cuando conectamos una puerta CMOS a una TTL y una TTL a otra CMOS.

Si conectamos una CMOS a una TTL nos encontramos que  $V_{oH}$  y  $V_{oL}$  de la puerta CMOS son ambas más pequeñas que los valores de  $V_{iH}$   $V_{iL}$  de la puerta TTL, por tanto no se nos puede presentar el caso de que la puerta TTL interprete como “0” un “1” ni de que interprete como “1” un “0”. Análogamente, si se conecta directamente una puerta TTL a otra puerta CMOS se verifica que  $V_{oL}$  de la puerta TTL es inferior al valor de  $V_{iL}$  de la puerta CMOS, luego no hay lugar a error en cuanto a la

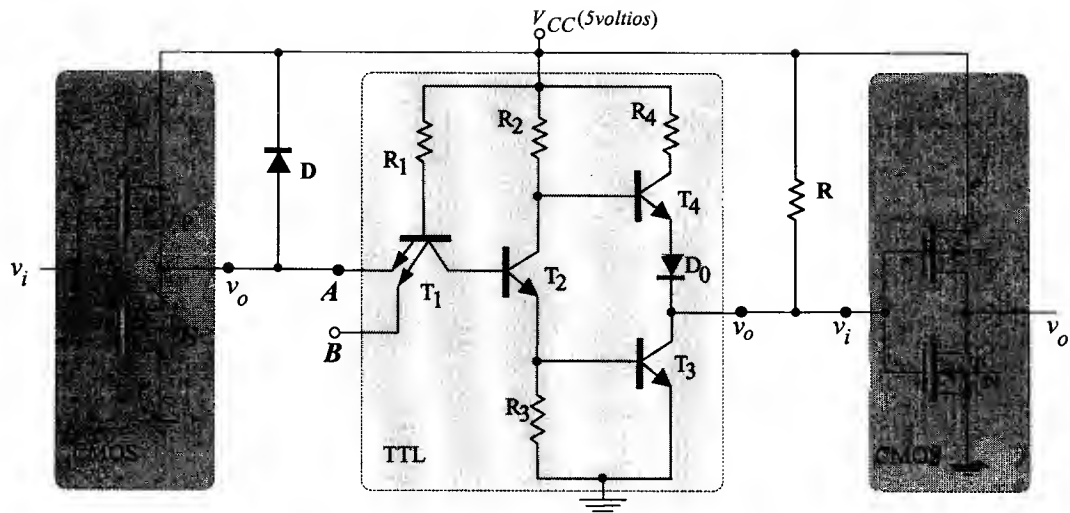
interpretación de un “0” a la entrada de la puerta CMOS. Sin embargo no ocurre lo mismo en el caso en el que la puerta TTL entrega un “1” a la puerta CMOS ya que  $V_{oH}$  de la puerta TTL es menor que  $V_{oL}$  de la CMOS. Es evidente que aquí necesitamos un desplazamiento de nivel. Esto se consigue conectando una resistencia entre la salida y la alimentación (5V) de la puerta TTL para así aumentar el valor de  $V_{oH}$  de la puerta TTL. El efecto de esta resistencia en paralelo con la resistencia de carga hace que ésta disminuya y por tanto en ella cae menos tensión cayendo más en el transistor de salida. Es decir, aumentando  $V_{oH}$ .



**Figura 4.25.** Valores típicos para las familias TTL y CMOS de los parámetros que caracterizan la viabilidad del acoplo entre estas familias.

Cuando se alimenta la puerta CMOS con una tensión mayor de 5V, conviene conectar a la salida de la puerta CMOS que se va a conectar con la puerta TTL un diodo para impedir que la señal de entrada a la puerta TTL supere los 5V. Este diodo se conecta entre la salida y la alimentación de forma que si la señal de entrada a la puerta TTL supera los 5V el diodo conduce e impide, actuando de retenedor, que a la entrada de la puerta TTL se le presenten más de 5V.

Cuando la puerta CMOS es una puerta de baja corriente, no es capaz de suministrar la corriente necesaria para conmutar las puertas TTL. La solución entonces es usar un Buffer CMOS que es capaz de absorber una corriente  $I_{oL}$  superior a la  $I_{iL}$  que le entrega la puerta TTL cuando está en baja y, además, es capaz de suministrar una  $I_{oH}$  suficiente a la puerta TTL cuando está en alta. La figura 4.26 muestra estos sencillos circuitos de acoplo entre CMOS y TTL.



**Figura 4.26.** Acoplo CMOS-TTL-CMOS.

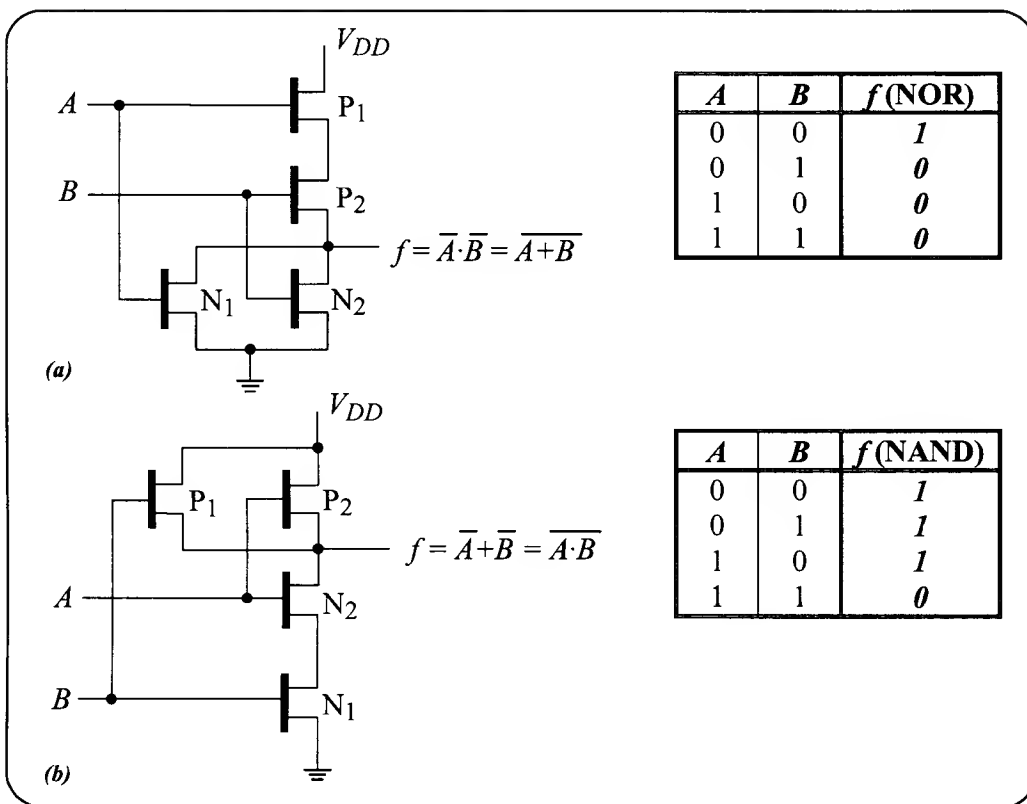
## 4.4. CIRCUITOS NAND Y NOR Y PUERTAS DE TRANSMISIÓN EN CMOS

Para comprender el principio de síntesis de las puertas básicas en CMOS basta recordar que cada entrada debe atacar a un transistor de canal N y otro de canal P. Si los de canal N están en paralelo y los de canal P en serie, obtenemos una puerta NOR y si los de canal P son los que están en paralelo y los de canal N en serie obtenemos una puerta NAND. La *figura 4.27* muestra los circuitos NOR y NAND en CMOS junto con sus tablas de verdad.

En la *figura 4.27.a* basta con que una de las entradas esté en alta para que el transistor  $N_1$  ó el  $N_2$  esté en saturación o en zona triodo y la salida sea baja. El par formado por los transistores  $P_1$  y  $N_1$  constituyen un inversor y el formado por  $P_2$  y  $N_2$  otro. El transistor  $P_2$  actúa como una resistencia en serie de valor muy alto o muy bajo dependiendo de la tensión de entrada. Análogamente el transistor  $P_1$  actúa como otra resistencia serie en el otro inversor,  $P_2$ - $N_2$ . Así, la salida sólo está en alta cuando los dos transistores de canal P,  $P_1$  y  $P_2$ , están en conducción. Esto sólo ocurre si ambas entradas están a tierra ("0").

Se pasa de NOR a NAND en CMOS intercambiando los transistores de canal P por canal N y dando la vuelta al circuito. La tabla de la *figura 4.28* muestra la forma de pasar de NAND a NOR y

viceversa realizando este tipo de intercambio en las posiciones relativas de los transistores de canal P y N.

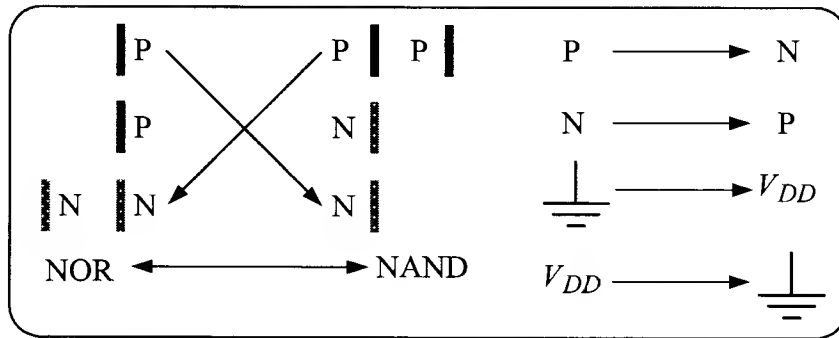


**Figura 4.27.** Puertas NOR y NAND en CMOS. (a) NOR. (b) NAND.

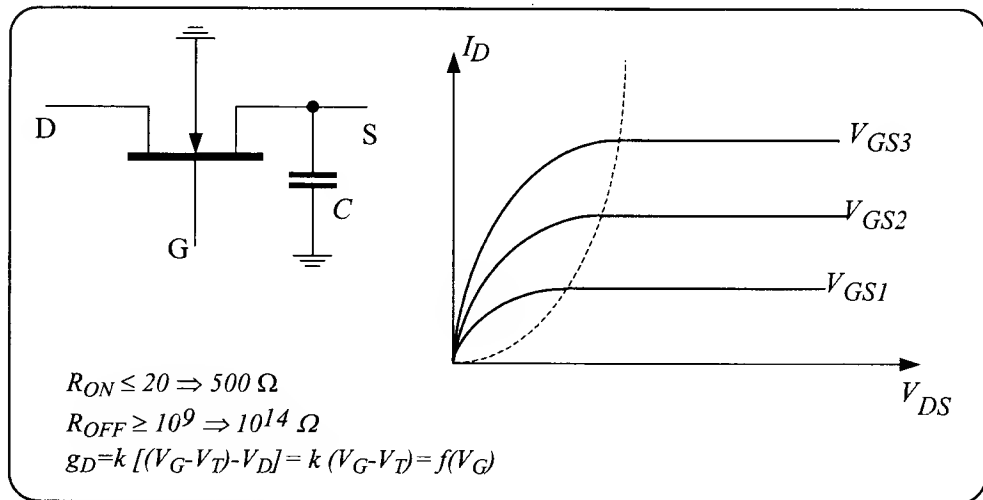
Otro bloque importante para la construcción de circuitos integrados en CMOS son las puertas de transmisión. El propio principio de funcionamiento de los MOS lleva implícita su aplicación a puertas de muestreo. La conductancia entre los terminales de fuente (S) y drenador (D) varía drásticamente en función del potencial en el otro terminal aislado (puerta, G) (figura 4.29).

Si la realizamos con un sólo transistor, idealmente no debería haber cambio en el nivel de señal entre drenador y fuente. Eso es cierto cuando  $V_D = 0V$ , pero cuando  $V_D = V_{DD}$ , con la puerta cerrada ( $V_G \gg V_T$ ), el transistor actúa como un seguidor por fuente y la capacidad  $C$  se carga a una tensión menor que  $V_D$  debido a  $V_T$ . Cuando el drenador cambia a "0", la capacidad se descarga a través del

transistor en sentido opuesto actuando este como una configuración en fuente común.



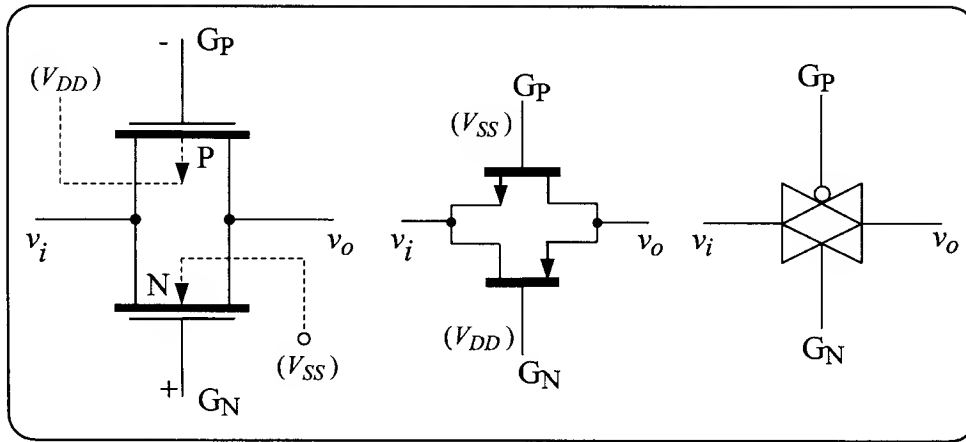
**Figura 4.28.** Reglas de transformación para pasar de NAND a NOR y viceversa cambiando la posición relativa de los transistores de canal P y N.



**Figura 4.29.** Puertas de transmisión en CMOS. Principio de funcionamiento. El transistor MOS es esencialmente un conmutador mandado por el potencial de puerta,  $V_G$ .

Usando transistores complementarios siempre habrá un transistor (canal P o N) que transmita la señal de la entrada a la salida sin atenuación significativa. Además, mejoran los transitorios ya que la capacidad parásita se carga y descarga siempre a través de un dispositivo en conducción. La puerta de

transmisión en CMOS es la de la *figura 4.30* en la que se incluyen también el esquema simplificado y el símbolo lógico.

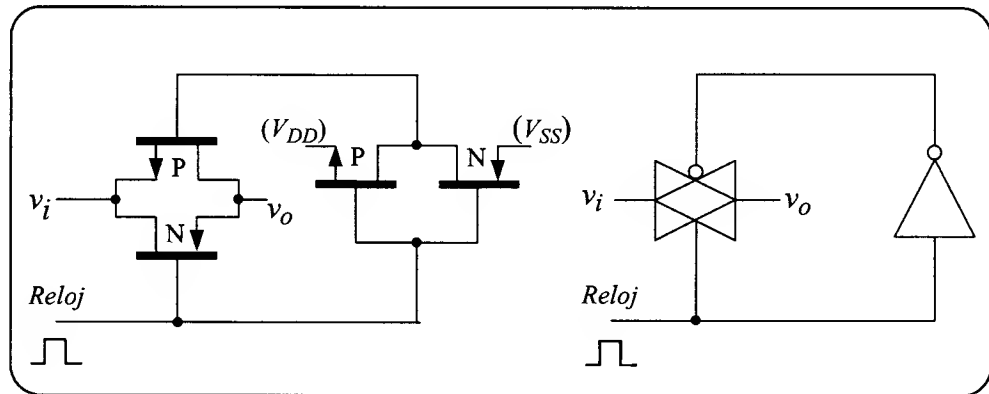


**Figura 4.30.** Circuito de la puerta de transmisión, esquema simplificado y símbolo lógico.

Cuando la puerta está en conducción la corriente puede pasar en ambas direcciones y esto ocurre cuando  $V_{GP}$  coincide con  $V_{SS}$ . La única limitación en el potencial de entrada es que debe encontrarse entre el potencial de drenador ( $V_{DD}$ ) y el del sustrato ( $V_{SS}$ ),  $V_{SS} < v_i < V_{DD}$ . Es decir, debe ser positivo respecto del sustrato ( $V_{SS}$ ) del dispositivo de canal N y negativo respecto del sustrato ( $V_{DD}$ ) del dispositivo de canal P. En el caso inverso, ( $V_{GP} = V_{DD}$  y  $V_{GN} = V_{SS}$ ), la puerta está en OFF ofreciendo un valor muy alto de impedancia ( $R > 10^9 \Omega$ ) y la entrada queda desconectada de la salida. Cuando se combina la puerta de transmisión en CMOS con un inversor se obtiene un conmutador bilateral, tal como se ilustra en la *figura 4.31*.

Las puertas de transmisión constituyen de hecho un bloque funcional básico para el diseño de toda la lógica estática, tanto combinacional como secuencial. Así, usando puertas de transmisión podemos sintetizar funciones lógicas elementales (OR, XOR, ...), circuitos de ruta controlada de datos (multiplexores y demultiplexores), circuitos aritméticos (sumadores), circuitos biestables tipo D, registros de desplazamiento, etc... Sin embargo, la necesidad de aumentar la densidad de integración y las prestaciones de los circuitos integrados en gran escala ha hecho que el diseño de puertas de transmisión se simplifique, abandonando la complementariedad de los procesos CMOS y buscando soluciones que sólo necesitan un transistor NMOS para sintetizar la función de un conmutador.

Veremos más adelante este uso de los conmutadores en la programación de las interconexiones en los dispositivos lógicos programables y en las celdas de memoria RAM dinámicas, entre otras aplicaciones.



**Figura 4.31.** Conmutador bilateral en CMOS

## 4.5. CIRCUITOS BiCMOS

Ya hemos comentado previamente que los inversores CMOS ofrecen unas características funcionales y estructurales muy próximas a las de un inversor ideal: alta impedancia de entrada, baja impedancia de salida, característica de transferencia simétrica con muy alta ganancia en la zona de transición, alta inmunidad al ruido, muy bajo consumo y, además, ocupando muy poca área de silicio, por lo que es posible alcanzar altas densidades de integración.

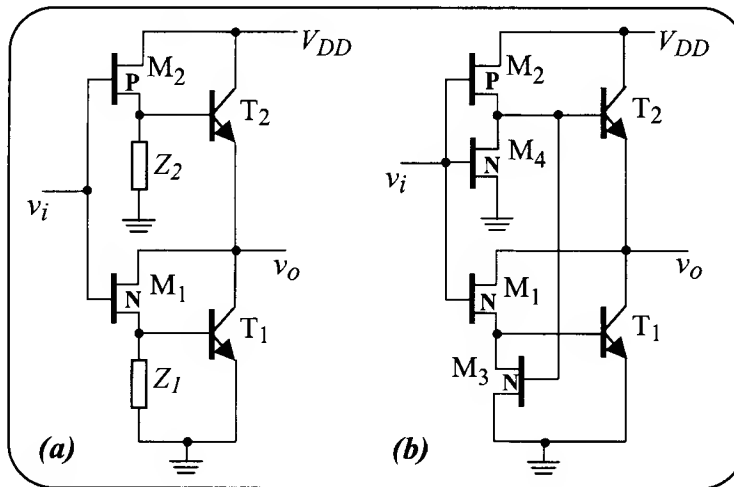
Sin embargo, CMOS tiene el inconveniente de la velocidad que siendo alta no alcanza los valores propios de la tecnología bipolar no saturada (ECL). La razón de esta limitación en la velocidad está en sus bajos niveles de corriente que no le permiten cargar y descargar capacidades tan rápido como ECL. Por eso se busca una solución de compromiso entre las buenas características de CMOS y la velocidad de ECL sin tener que pagar el alto precio del excesivo consumo de la lógica bipolar no saturada.

La solución salomónica es la de la tecnología BiCMOS que usa los procesos CMOS para la síntesis lógica inicial, manteniendo así la alta densidad de integración y el alto valor de la impedancia

de entrada, y termina con una configuración bipolar (tipo “push-pull”) capaz de suministrar los niveles de corriente necesarios para mover la carga almacenada en las capacidades rápidamente.

La *figura 4.32.a* muestra el esquema genérico de un inversor BiCMOS. Cuando la entrada está en alta, el transistor NMOS,  $M_1$ , está en conducción, haciendo que también conduzca el transistor bipolar  $T_1$ , mientras que  $M_2$  y  $T_2$  están al corte. Al conducir  $T_1$ , la tensión de salida,  $v_o$ , está en baja, ya que corresponde a la tensión colector-emisor de un transistor bipolar próximo a la saturación.

De forma complementaria, cuando la tensión de entrada,  $v_i$ , está en baja, conducen  $M_2$  y  $T_2$  y  $M_1$  y  $T_1$  están al corte. Así, la tensión de salida,  $v_o$ , es alta. Obsérvese que los dos transistores de salida,  $T_1$  y  $T_2$  actúan como un divisor de la tensión de alimentación,  $V_{DD}$ , en dos partes, dependiendo del valor de las impedancias de  $T_1$  y  $T_2$  cuando están al corte y cuando conducen. Ahora, cuando  $T_1$  está al corte, su impedancia equivalente (unión de colector polarizada en inversa) es mucho mayor que la de  $T_2$ , por lo que prácticamente toda la tensión  $V_{DD}$  cae entre el colector de  $T_1$  y tierra y, por consiguiente, la tensión  $v_o$  es la del estado de alta.



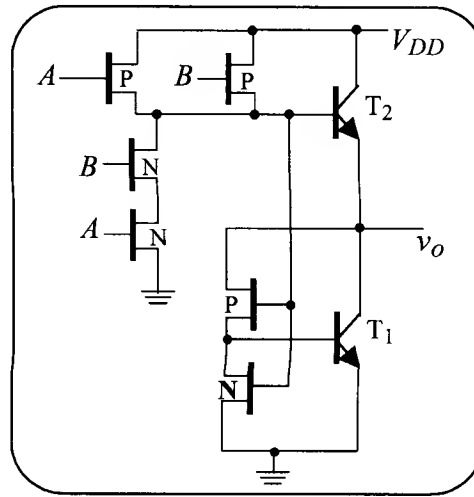
**Figura 4.32.** (a) Esquema básico de un inversor BiCMOS. (b) Inversor BiCMOS donde se han sustituido las impedancias  $Z_1$  y  $Z_2$  por los transistores  $M_3$  y  $M_4$ .

Para mantener la función de las impedancias  $Z_1$  y  $Z_2$ , necesarias para extraer la carga de base de los transistores bipolares cuando pasan al corte y para facilitar la integración se sustituyen por otros dos transistores, tal como se muestra en la *figura 4.32.b*. Estos nuevos transistores,  $M_3$  y  $M_4$ , poseen



impedancias activas equivalentes a  $Z_1$  y  $Z_2$  y, además, cuando no son necesarios no están activos y, por consiguiente, no consumen.

Una vez captada la idea del diseño BiCMOS (MOS en la entrada y etapa bipolar push-pull a la salida), es fácil pasar del inversor a otras puertas NAND o NOR. Por ejemplo, la *figura 4.33*. muestra el circuito correspondiente a una puerta NAND. Obsérvese que no hemos tocado la parte de salida y sólo hemos cambiado la parte CMOS de entrada.



**Figura 4.33.** Puertas NAND de 2 entradas en BiCMOS.

## 4.6. COMPARACIÓN DE FAMILIAS LÓGICAS

Dado el propósito esencialmente pedagógico de este texto y la rápida evolución de la tecnología electrónica es difícil y de dudosa utilidad el intentar comparar series concretas dentro de una familia y entre familias. Por eso el contenido de este apartado es de naturaleza cualitativa. La tabla de la *figura 4.34* muestra un resumen de las ventajas e inconvenientes de las distintas familias.

Desde su introducción a comienzos de los años sesenta (serie 4000), la familia CMOS ha ido evolucionando, incorporando distintas mejoras en el proceso de fabricación para optimizar sus características propias (bajo consumo, alta inmunidad al ruido y diversidad de fuentes de alimentación) y mejorar en otras características inicialmente asociadas a otras familias, tales como la

velocidad de conmutación. Las series 4000B y 74C de CMOS con puerta de silicio redujeron el retardo de propagación y el área ocupada, con un consumo del orden de los  $10nW$ , en comparación, por ejemplo, con los  $2mW$  de las series TTL de bajo consumo. El comportamiento de la serie HCMOS ( $9ns$  de tiempo de propagación) es comparable en velocidad y corriente disponible a las series 54LS y 74 LS, pero con un consumo mucho menor.

Familia	Ventajas	Inconvenientes	Otras familias mejoradas
TTL (74xx)	<ul style="list-style-type: none"> <li>♦ <i>El menor producto retardo por disipación de potencia</i></li> <li>♦ Buena flexibilidad lógica</li> <li>♦ Baja impedancia de salida</li> <li>♦ Buena inmunidad al ruido</li> <li>♦ Numerosas funciones</li> </ul>	<ul style="list-style-type: none"> <li>♦ Generación de ruido</li> </ul>	74Hxx 74Sxx 74Lxx 74LSxx 74ASxx 74ALSxx
ECL	<ul style="list-style-type: none"> <li>♦ <i>El menor retardo de propagación</i></li> <li>♦ Buena flexibilidad lógica</li> <li>♦ Salidas complementarias</li> <li>♦ Baja impedancia de salida</li> <li>♦ Buena inmunidad al ruido</li> <li>♦ Baja generación de ruido</li> </ul>	<ul style="list-style-type: none"> <li>♦ Alta disipación (40 mW)</li> <li>♦ Necesita circuito de adaptación con otras familias</li> </ul>	MECL 10K MECL 100K MECL 300K
MOS	<ul style="list-style-type: none"> <li>♦ Alto fan-out</li> <li>♦ Gran densidad de integración</li> <li>♦ La NMOS es más rápida que la PMOS</li> </ul>	<ul style="list-style-type: none"> <li>♦ Incompatibilidad con otras familias</li> <li>♦ Alta impedancia de salida</li> <li>♦ No admite cableado lógico</li> <li>♦ Baja velocidad (PMOS)</li> </ul>	
CMOS (400Cxx, 74Cxx)	<ul style="list-style-type: none"> <li>♦ <i>La de menor disipación de potencia</i></li> <li>♦ Amplios márgenes de ruido</li> <li>♦ Alto fan-out y alto fan-in</li> <li>♦ Amplios márgenes en la alimentación</li> <li>♦ Buena inmunidad al ruido</li> </ul>	<ul style="list-style-type: none"> <li>♦ No admite cableado lógico</li> <li>♦ Menos rápida que TTL y ECL</li> </ul>	74HCxx: 74HCTxx 74ACTxx 74AHCTxx 74FCTxx 74ACTQxx 74VHCxx 74VHCTxx

H: Alta velocidad. L: Baja potencia. S: Schottky. A: Avanzada. C: CMOS. T: Compatible con TTL. V: muy., U: Ultra.  
 FAST™ t FACT™: Marca registrada Fairchild: Circuitos avanzados de alta velocidad en TTL y CMOS.  
 TingLogic™ de Fairchild: HS, HST, UHS en tecnología CMOS (2 micrón) que admite alimentación variable  
 Crossvolt™ de Fairchild: LCX, LVX VCX en tecnología CMOS y LVT y LVTH en tecnología BICMOS

**Figura 4.34.** Tabla de comparación entre las distintas familias lógicas. Las familias mejoradas se basan en las básicas correspondientes en las que se ha optimizado alguno de sus parámetros.

Hemos visto también cómo la familia BiCMOS intentaba quedarse con lo mejor de CMOS y ECL, añadiendo una etapa bipolar a la salida para aumentar la capacidad de las puertas para manejar niveles de corriente más altos que los usuales en CMOS.

En resumen, en cuanto a consumo la mejor solución es CMOS a  $5V$ . Si lo esencial es el tiempo de propagación, debe elegirse ECL, pero recordando que su consumo es del orden de  $25mW$ , frente a los  $5\mu W$  de CMOS. En cuanto a inmunidad al ruido la mejor es CMOS a  $18V$  (si bien aquí no hemos considerado la familia de alto umbral, HTL, que sería la vencedora en este parámetro). Cuando aumenta el nivel de integración, la balanza se inclina de nuevo hacia MOS y CMOS.

Conviene recordar finalmente que el campo propio de la tecnología CMOS es la integración en muy gran escala donde el conjunto de sus características (alta densidad, alta velocidad, bajo consumo y bajo coste de fabricación) la han convertido en dominante en el estado actual de la tecnología electrónica. Tendremos ocasión de comprobarlo a lo largo del texto, a medida que aumente la complejidad de los circuitos. Por ejemplo, en las memorias EEPROM reprogramables eléctricamente, en las memorias RAM estáticas y dinámicas, en los circuitos lógicos programables, etc. En todos los casos, sea cual fuere la arquitectura y la función, el proceso de base es CMOS.

## 4.7. PROBLEMAS<sup>(\*)</sup>

Hemos visto las familias lógicas ECL, MOS y CMOS siguiendo el mismo procedimiento del tema anterior. Es decir, dentro de cada familia hemos elegido una puerta representativa, NAND, NOR o inversor y hemos explicado su funcionamiento. En los casos MOS y CMOS los circuitos son más sencillos que en ECL. En estos ejercicios intentamos completar el contenido proponiendo el análisis de circuitos análogos a los incluidos en el tema.

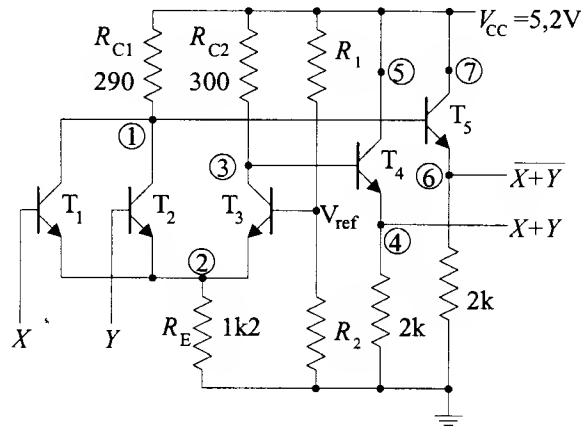
**E.4.1.** *Con ayuda de las figuras y los comentarios referentes a la puerta NOR de la familia ECL del texto realizar la simulación del circuito de la figura 4.3 y obtener su característica de transferencia para comprobar que realmente se comporta como un inversor. Determinar con ayuda del simulador la función de los transistores  $T_3$  y  $T_4$ . ¿Por qué son necesarios?.*

---

<sup>(\*)</sup> La solución de estos ejercicios se encuentra en el texto de problemas.

¿En qué región trabajan?, ¿por qué?. Compárense las salidas inversoras y no inversoras con las tensiones en los nudos 1 y 2. ¿Cuál es la ventaja de disponer de salida inversora y no inversora?

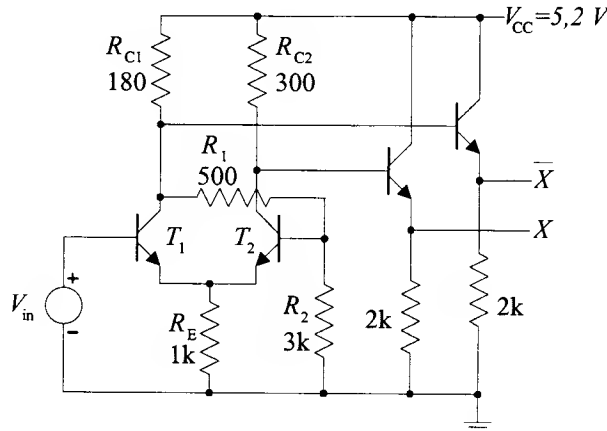
- E.4.2.** La figura adjunta muestra el esquema mínimo de una puerta NOR en ECL, en el que, además, se ha separado el amplificador diferencial de entrada ( $T_1$ ,  $T_2$ ,  $T_3$ ) de los dos transistores seguidores de emisor ( $T_4$  y  $T_5$ ) usados para desplazar los niveles de tensión y modificar las impedancias de salida.



- Analizar primero el par diferencial y comprobar cualitativamente que en los puntos 1 y 3 aparecen las funciones NOR y OR de las entradas ( $X$ ,  $Y$ ).
  - Calcular  $R_1$  y  $R_2$  para que el valor de la tensión de referencia sea  $1.2V$ .
  - Analizar el comportamiento de  $T_4$  y  $T_5$ . ¿Invierten la señal?. ¿Cuál es la diferencia entre el potencial en 3 y en 4?. ¿Por qué?.
  - Sabiendo que  $V_{CC} = 5.2V$  y  $V_{ref} = 1.2V$ , estimar los valores de las tensiones en los puntos 1, 2, 3, 4, 5, 6, y 7 cuando entra  $X="1"$ ,  $Y="0"$  y cuando entra  $X=Y="0"$ . ¿Qué valores considera razonables para  $V("0")$  y  $V("1")$  sabiendo que  $V_{ref} = 1.2V$  y que  $V_{BE(ON)} = 0.7V$ ,  $V_{CE(sat)} = 0.2V$  y  $V_{BE(corte)} = 0.5V$ .
- E.4.3.** Explicar el funcionamiento de los circuitos de acoplo entre ECL y TTL que se muestran en las figuras 4.6 y 4.7.a. Resolver el divisor de tensión de la figura 4.6 (con  $R_1=180\Omega$ ,  $R_2=270\Omega$  y  $R_3=820\Omega$ ) usado para atenuar las salidas TTL hasta los niveles que requiere ECL. En la explicación del funcionamiento del circuito de la figura 4.7.a conviene darse

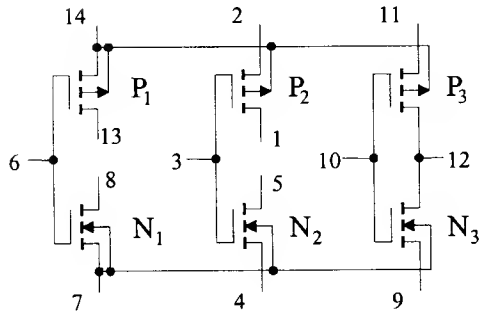
cuenta que los transistores usados en el par diferencial son PNP.

- E.4.4.** Dado el circuito de la figura dibujar la característica de transferencia cuando la entrada pasa de estado bajo a estado alto. Vuélvase a calcular ahora pero suponiendo que la entrada pasa de estado alto a estado bajo. Discútanse los resultados obtenidos. ¿Para qué tipo de aplicaciones puede resultar práctica esta puerta?. Obténgase la característica de transferencia por simulación. Supónganse  $V("0") = 3.5V$  y  $V("1") = 4.5V$ .



- E.4.5.** La figura 4.28 del texto muestra las reglas de transformación para puertas CMOS NAND y NOR. Aplicalas proponiendo las soluciones complementarias a las de la figura 4.27.
- E.4.6.** La figura 4.12.b muestra las características de transferencia del inversor CMOS. Compruebe que las comprende contestando a las siguientes cuestiones:
- ¿Cuál es el significado físico de las tensiones  $V'_{TP}$  y  $V_{TN}$ ?
  - En la característica hay marcadas cinco regiones que corresponden a distintas configuraciones en los estados de los transistores  $T_p$  y  $T_n$  (corte, saturación, triodo). Compruebe la tabla de la figura 4.22 mostrando la configuración correspondiente a cada una de esas regiones. Usar como apoyo las figuras 4.20 y 4.21.a.
- E.4.7.** El circuito integrado 4007UB de la figura adjunta contiene tres transistores canal P y tres de canal N trabajando en modo de realce y con sus terminales accesibles desde el exterior para poder configurar al circuito en la realización de distintas funciones. ¿Cómo deberían realizarse las conexiones externas para obtener:

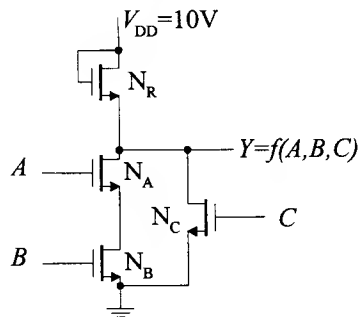
- Tres inversores independientes.
- Una puerta NOR de 3 entradas.
- Una puerta NAND de 3 entradas.
- Una puerta de transmisión bidireccional.
- Un inversor de tres estados con la siguiente tabla de control:



Entrada	Facilitación / Inhibición	Salida
1	0	0
0	0	1
X	1	<b>abierta</b>

**E.4.8.** Realícese la simulación del inversor CMOS y obténgase la característica de transferencia y la corriente  $I_D = I_D(V_G)$ . Calcúlense gráficamente las tensiones umbrales de los dos transistores que intervienen ( $V_{TN}$ ,  $V_{TP}$ ) y los márgenes de ruido a partir de los datos obtenidos en simulación.

**E.4.9.** Dado el circuito lógico de la figura adjunta determinar la función lógica que realiza. Comprobar los razonamientos mediante la simulación.



---

# Preparación de la Evaluación

---

## Objetivos

- 1: Familia Lógica ECL
- 2: Familia Lógica MOS
- 3: Familia Lógica CMOS
- 4: Acoplo y comparación de familias

### 1. Familia Lógica ECL

- 1.1. *¿Podría describir las dificultades de la familia TTL para aumentar la velocidad de conmutación y las razones por las que ECL no tiene ese problema?*
- 1.2. *La figura 4.4.de este tema muestra las características de transferencia de un circuitos ECL (serie 10K). ¿Podría explicarla y calcular sus márgenes de ruido para los niveles alto y bajo?. ¿Qué reflexión le merece esos resultados en comparación con los de TTL?*
- 1.3. *En el esquema básico de ECL explicado en el tema hemos hecho énfasis en el amplificador diferencial de entrada (transistores  $T_1$  y  $T_2$ ), ¿podría ahora explicar la función que realizan los transistores  $T_3$  y  $T_4$  y los cambios que introducen en el valor de corrientes y tensiones?*
- 1.4. *El circuito de la figura adjunta es prácticamente idéntico al que hemos usado en el tema pero con valores diferentes en las resistencias. Intente calcular de forma aproximada el valor de los niveles lógicos,  $v("0")$  y  $v("1")$ .*

Recuerde que:  $V_{BE}(T_1)$  en activa  $\approx 0.7V$

$$I_E("0") = \frac{-5.2 - (V_B + V_{BE})}{R_B}$$

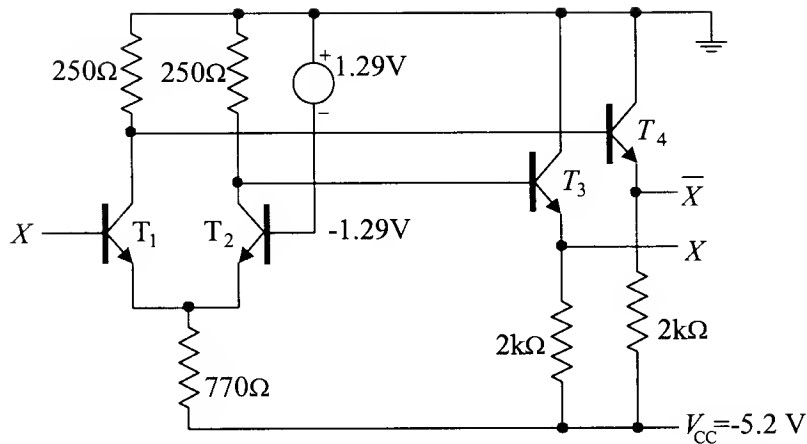
$$I_E("1") = \frac{-5.2 - (v_i + V_{BE})}{R_E}$$

$$V_{C2} = I_{C2} \cdot R_{C2} \approx I_E("0") \cdot R_{C2}$$

$$V_{C1} = I_{C1} \cdot R_{C1} \approx I_E("1") \cdot R_{C1}$$

$$v_o("0") = V_{C2} - V_{BE}(T_3)$$

$$v_o("1") = V_{C1} - V_{BE}(T_3)$$



## 2. Familia Lógica MOS

- 2.1. ¿Qué diferencias esenciales hay entre las tecnologías Bipolar y MOS?. ¿Por qué es más adecuada la segunda para alta densidad de integración?.
- 2.2. ¿Qué analogías y diferencias encuentra entre RTL y PMOS?. ¿Podría explicar por qué un MOS de carga y canal N opera como una resistencia al unir drenador y fuente?.
- 2.3. Intente explicar el significado físico de la curva de la figura 4.13.
- 2.4. Recordando la función de resistencia que realiza el transistor MOS de carga, ¿podría dibujar los circuitos correspondientes a puertas NAND y NOR de tres entradas?.

## 3. Familia Lógica CMOS

- 3.1. ¿Podría enumerar las características de una familia lógica ideal?. ¿Podría compararlas con las de la familia CMOS?.
- 3.2. Hemos visto en el tema el estudio detallado del inversor CMOS con dos transistores MOS trabajando en modo de realce, uno de canal N ( $T_N$ ) y otro de canal P ( $T_P$ ) con los niveles lógicos:

$$v("1") = V_{DD}$$

$$v("0") = V_{SS} = 0V$$

¿Podría explicar la característica de transferencia, y obtener los posibles puntos de trabajo para un inversor CMOS en el que el transistor impulsor es de canal P y la carga de canal N?.



*¿En qué se modifican los niveles lógicos?*

- 3.3. *La tabla 4.2 resume el estado funcional de  $T_N$  y  $T_P$  en cada una de las cinco zonas de la característica de transferencia. ¿Podría explicarla describiendo cualitativamente los procesos que ocurren a medida que la tensión de entrada,  $v_i$ , pasa desde 0 hasta  $V_{DD}$ ?*
- 3.4. *Use el simulador para obtener las características de transferencia del inversor CMOS. Si encuentra dificultades, le sugerimos que revise el problema resuelto E.4.8. Allí encontrará “pistas” y “buenos valores” de tensión y de los incrementos.*
- 3.5. *Dibuje el circuito correspondiente a las puertas NAND (NOR) en CMOS y use las reglas de transformación de la figura 4.22 para pasar a NOR (NAND).*
- 3.6. *Comente la idea que subyace a todas las puertas de tres estados. Extienda la estrategia seguida para convertir el inversor CMOS a tres estados a las otras puertas (NAND y NOR). Recuerde que lo que buscamos siempre es la posibilidad de aislar el punto de salida,  $v_o$ , tanto de la fuente,  $V_{DD}$ , como de la tierra.*

#### **4. Acoplo y comparación de familias.**

- 4.1. *¿Podría enumerar y definir los criterios de evaluación de familias descritos al comienzo del tema 3?. Aplíquelos a ECL y CMOS.*
- 4.2. *Explique el funcionamiento de los circuitos de acoplo entre ECL y TTL que se muestran en las figuras 4.6 y 4.7 del texto. (Acuda al problema resuelto E.4.3 si encuentra dificultades).*
- 4.3. *¿Podría explicar la necesidad del acoplo entre circuitos pertenecientes a las familias lógicas TTL y CMOS?. ¿Y entre TTL y ECL?. ¿Y entre ECL y CMOS?.*



## REFERENCIAS BIBLIOGRÁFICAS

1. Carmona Suárez, E., Mira Mira, J.: “*Problemas Resueltos de Física de Dispositivos Electrónicos*”. UNED (ref. 07407). Madrid, 2000
2. Mira, J., Delgado, A.E.: “*Electrónica Analógica Lineal*” Tomos 1. UNED (ref. 074076). Madrid, 1993.
3. Rabaey, J.M.: “*Digital Integrated Circuits. A Design Perspective*”. Prentice Hall Inc. 1996
4. Uyemura, J.P.: CMOS Logic Circuit Design. Kluwer Academic Pub. Mass. 1999.
5. Weste, N.H.E. y Eshraghian, K.: “*Principles of CMOS VLSI Design*”. Addison–Wesley Pub., 1994.



---

## ***Lógica Combinacional (I): Funciones Aritmético-Lógicas***

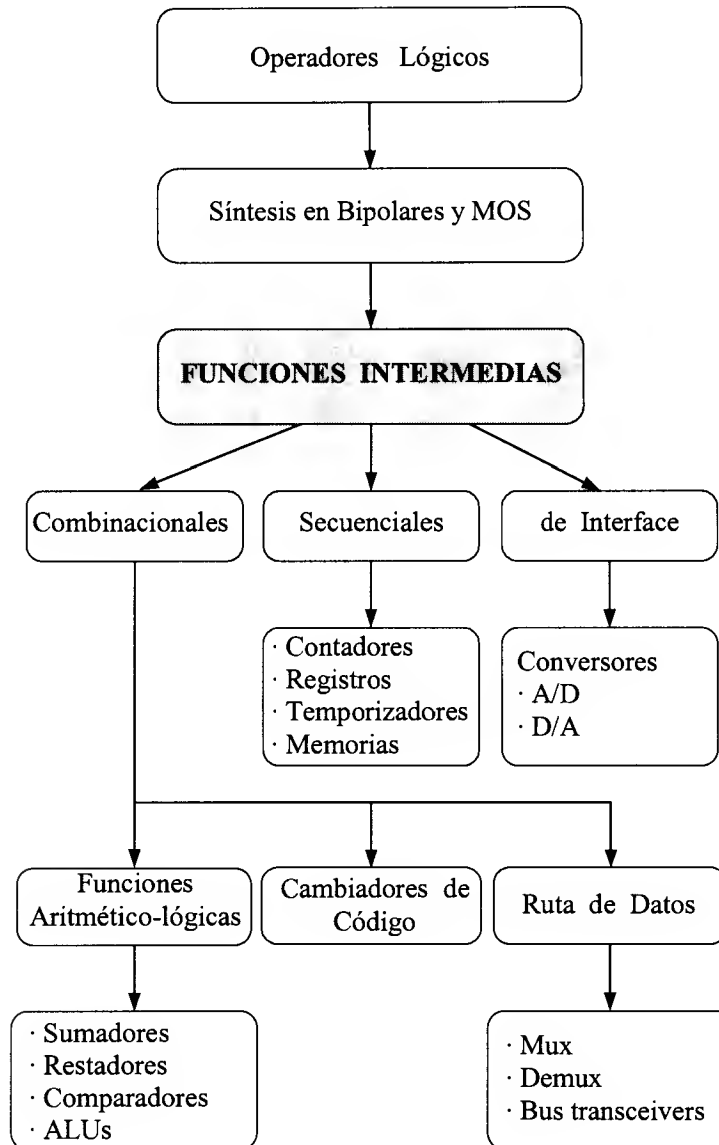
---

### **+ CONTEXTO**

*En los temas anteriores hemos visto los operadores básicos necesarios para la síntesis de funciones en electrónica digital, las bases electrónicas necesarias para comprender el diseño de estos operadores (transistores y diodos en corte y saturación) y las distintas familias lógicas que aportan soluciones concretas en tecnologías bipolar y MOS.*

*A partir de aquí podemos volver al nivel lógico y considerar las puertas AND, OR, NOT, NAND y NOR como dispositivos de varios terminales caracterizados por su tabla de verdad y por los parámetros de la familia correspondiente. Es decir, como módulos funcionales de bajo nivel a partir de los cuales se diseñan los módulos del nivel intermedio, sin preocuparnos en general de su estructura interna.*

*Empezamos así la siguiente etapa de la electrónica digital ("sistemas digitales"), centrada en torno al análisis de necesidades computacionales de los grandes sistemas de procesamiento digital de la información y al desglose de esas necesidades en un nivel de funciones intermedias que se corresponden aproximadamente con el nivel de integración en media escala. La figura adjunta sitúa en contexto estas funciones.*



*En general, estas funciones se realizan a nivel de registro entre palabras de  $n$  bits y son la base de todos los sistemas digitales de comunicación, instrumentación, cálculo y control. Toda computación digital termina siendo una secuencia de operaciones aritmético-lógicas sobre un conjunto de palabras de  $n$  bits.*

*Hay esencialmente tres tipos de funciones en lógica combinacional:*

- a) Funciones aritmético-lógicas (ALU)*
  - b) Funciones de ruta de datos*
  - c) Circuitos cambiadores de código*
- 
- a) Funciones aritmético-lógicas, (sumadores, restadores, multiplicadores y operaciones lógicas bit a bit). Son las encargadas de realizar operaciones locales entre dos datos de  $n$  bits.*
  - b) Funciones de ruta de datos, (multiplexos, demultiplexos, transmisores-receptores de bus, etc...). Su función es guiar el tráfico de datos e instrucciones entre las distintas partes de un sistema de cálculo (de memoria a unidad aritmética, etc...). La clave de estas funciones es el carácter controlado del movimiento a través de puertas que se abren o cierran de forma síncrona, en general, de acuerdo con los pulsos de un reloj.*
  - c) Circuitos cambiadores de código. Para cada tipo de proceso existe una representación digital de la información que es más adecuada que otras. Por eso es necesario disponer de circuitos cambiadores de código. Su ejemplo más general es el de las memorias de sólo lectura (ROM), que son en realidad circuitos que sintetizan funciones múltiples de forma que cada bit de la palabra de salida puede ser una función lógica cualquiera de todos los bits de entrada.*

*Las funciones secuenciales las estudiaremos más tarde y las de interfaz (esencialmente los conversores analógicos digitales y digitales analógicos) no las estudiamos en este texto.*

*En este tema estudiaremos las operaciones aritmético-lógicas, los comparadores y la estructura interna de una ALU ("Aritmético-Logical Unit"). En el capítulo siguiente estudiaremos las funciones de ruta de datos y después, la lógica combinacional programable.*

*Por razones pedagógicas, estudiaremos la estructura interna de los circuitos sumadores y*

*restadores, de los comparadores y de las ALU's. Sin embargo, el alumno debe recordar que estas funciones ya existen integradas en media escala (SN74283, sumador binario paralelo de 4 bits; SN74181, ALU; SN74182, generador de acarreo adelantado de 32 bits; SN74281, multiplicador binario paralelo de 2 bits x 4 bits) y como componentes reutilizables en una biblioteca de motivos para ser usados en el diseño asistido por computador de otras funciones más complejas.*

*Aquellos alumnos de informática que estudien este texto de Electrónica Digital a la vez que cursan otras asignaturas como Estructura y Tecnología de Computadores encontrarán cierto grado de solape. Aquí hacemos énfasis en el diseño electrónico. Allí se extiende el estudio de la aritmética incorporando versiones más rápidas, multiplicadores y aritmética en coma flotante pero haciendo más énfasis en la descripción algorítmica y con diagramas de bloques y en el papel concreto que juegan estos circuitos en la arquitectura de un computador.*

## **++ CONOCIMIENTO PREVIO NECESARIO**

*Todos los conocimientos previos necesarios sobre la representación, análisis, minimización y síntesis de funciones lógicas se encuentran en el tema 1, de modo que en principio no deberíamos tener problemas al estudiar los circuitos necesarios para realizar operaciones aritméticas y lógicas con palabras de  $n$  bits.*

*Esto es cierto si sólo tuviéramos que operar con magnitudes enteras y positivas y si no tuviéramos limitaciones en cuanto a la longitud de los registros y la capacidad de memoria de un procesador digital. Sin embargo, necesitamos representar también los números enteros negativos y el rango de valores posibles es limitado. Por eso introducimos un apartado inicial en el tema dedicado al estudio de los distintos métodos de representación conjunta de números positivos y negativos.*

## **+++ OBJETIVOS DEL TEMA**

*El objetivo general de este capítulo es dar a conocer la estructura interna de los circuitos que son capaces de realizar las operaciones aritméticas y lógicas necesarias para sintetizar las ALU's de cualquier sistema de cálculo digital. Por consiguiente, comienza con la descripción*

*formal o algorítmica de una operación, busca su síntesis eficiente usando puertas NAND o NOR y, finalmente, conecta con la electrónica real haciendo referencia a las soluciones MSI (integración en media escala) para esas funciones. Estas funciones integradas son (sumadores-restadores, ALU's y multiplicadores) a su vez componentes reutilizables de una biblioteca de módulos software para el diseño de otras funciones más complejas en LSI y VLSI.*

**Objetivo 1:** *Conocer los fundamentos de la aritmética binaria. Distintas formas de representación de los números positivos y negativos.*

**Objetivo 2:** *Conocer los distintos tipos de circuitos sumadores y restadores (soluciones básicas y rápidas y problema del rebose).*

**Objetivo 3:** *Comprender el funcionamiento de los comparadores para palabras de  $n$  bits.*

**Objetivo 4:** *Comprender el funcionamiento y la estructura interna de las ALU's tipo SN74181.*

## **++++ GUÍA DE ESTUDIO**

*El conocimiento de este tema es muy clásico y se encuentra, con muy pocas variaciones, en todas las referencias complementarias. También se encuentra en la mayoría de los textos de Estructura y Tecnología de Computadores como introducción al estudio de las unidades aritmético-lógicas más completas y complejas que el circuito SN74181, tomado aquí como límite.*

*La estrategia seguida en la presentación del material es estudiar primero la función para un bit, pasar después a operandos de  $n$  bits y, finalmente, estudiar los circuitos que aumentan la velocidad y/o la precisión de la operación. Aquí estudiaremos sólo el aumento de la velocidad buscando soluciones rápidas al problema del arrastre. No haremos mención a la aritmética en coma flotante.*



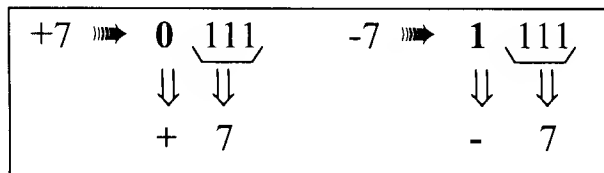
---

## Contenido del Tema

---

### 5.1. REPRESENTACIÓN CONJUNTA DE NÚMEROS POSITIVOS Y NEGATIVOS

Tal como comentábamos en el apartado sobre conocimiento previo necesario para este tema es cierto que para operar con magnitudes enteras y positivas ya disponemos de todos los conocimientos previos necesarios para analizar y diseñar circuitos que realicen operaciones aritméticas y lógicas con palabras de  $n$  bits. Sin embargo, necesitamos representar también los números enteros negativos y el rango de valores posibles es limitado. Una palabra de  $n$  bits sólo admite  $2^n$  configuraciones diferentes y de estas, la mitad se gasta en los enteros positivos, una se gasta con el cero y el resto tenemos que gastarlo para representar los números negativos. Así, por ejemplo, si nos limitamos a una máquina que trabaje con palabras de 4 bits, el máximo entero positivo (1111) vale 15 y no pueden existir operandos ni resultados de operaciones que superen ese valor. Si queremos incluir números negativos, tendremos que reservar un bit para el signo, con lo que el máximo número decimal entero es ahora el que corresponde a los tres bits restantes en alta (111). Es decir:



En general, para palabras de  $n$  bits, tendremos  $2^{n-1}$  números positivos y  $2^{n-1}$  negativos.

Hay tres formas básicas que permiten la representación conjunta de números positivos y negativos: (1) *Signo y Magnitud (S-M)*, (2) *Complemento a 1 (C-1)* y (3)



*Complemento a 2 (C-2)*. Obviamente, la forma de representación define el circuito que sintetiza el sumador correspondiente.

Las tres formas de representación usan el bit más significativo para codificar el signo del número. Los números que empiezan por cero son positivos y los que empiezan por uno son negativos. El resto de los bits codifican su magnitud. Por consiguiente, en el formato *Signo-Magnitud (S-M)* la codificación es inmediata. Primero se pone el signo (0 = *positivo*, 1 = *negativo*), y después los bits de magnitud. Así, negar un número es simplemente reemplazar el bit de signo por su complementario. Obsérvese que aquí el número cero tiene una doble representación (+0 = 0000 y -0 = 1000, para 4 bits).

Para sumar dos números positivos o dos números negativos, sumamos sus bits de magnitud y asignamos al resultado el signo de los sumandos. Así por ejemplo,

$$0\ 100\ (+4) + 0\ 010\ (+2) = 0\ 110\ (+6)$$

$$1\ 100\ (-4) + 1\ 010\ (-2) = 1\ 110\ (-6)$$

Cuando los signos de los operandos no coinciden, restamos el de menor magnitud del otro y el resultado hereda el signo del de mayor magnitud. Así, por ejemplo:

$$\underline{0}\ 100\ (+4) + \underline{1}\ 010\ (-2) = \underline{0}\ 010\ (+2)$$

$$\underline{1}\ 100\ (-4) + \underline{0}\ 010\ (+2) = \underline{1}\ 010\ (-2)$$

Por eso se han buscado otras formas de representación, porque la que es aparentemente más sencilla obliga a una síntesis más complicada. Si un circuito sumador tiene que manejar números negativos en la representación signo-magnitud necesitará incluir, además del sumador, un restador y dos comparadores, uno para el signo y otro para la magnitud.

En la representación en *Complemento a 1 (C-1)*, los números negativos se obtienen complementando uno a uno todos los bits de la representación del número positivo correspondiente. Así +3 =  $\underline{0}\ 011$  y -3 =  $\underline{1}\ 100$ .

Obsérvese que también aquí tenemos dos representaciones diferentes para el cero (+0 =  $\underline{0}\ 000$  y -0 =  $\underline{1}\ 111$ ), pero la ventaja de esta representación es que la resta se implementa con una negación y una suma:  $A-B=A+(-B)$  Para eliminar la complicación asociada a la doble representación del cero se usa el método del complemento a la base, o sea complemento

a 2 (**C-2**).

En la representación en *complemento a 2 (C-2)*, los números positivos se representan colocando un cero a la izquierda de los bits que representan la magnitud, como en las otras dos representaciones. Los números negativos se representan a partir de los positivos correspondientes complementando todos los bits y sumando un "1" al resultado. Al igual que en la representación en *C-1*, la resta se implementa con negación y suma, pero ahora el cero sólo tiene una representación ( $+0 = 0\ 000$ ). Para comparar las *tres representaciones* conviene observar las tablas de las *figuras 5.1* y *5.2* viendo las analogías y diferencias para palabras de cuatro bits (tres de magnitud y uno de signo).

<b>Configuraciones Binarias (4 bits)</b>	<b>Nº equivalente en decimal según las distintas representaciones en binario</b>			
	<b>Binario Puro</b>	<b>S-M</b>	<b>C-1</b>	<b>C-2</b>
0 000	0	+0	+0	+0
0 001	1	+1	+1	+1
0 010	2	+2	+2	+2
0 011	3	+3	+3	+3
0 100	4	+4	+4	+4
0 101	5	+5	+5	+5
0 110	6	+6	+6	+6
0 111	7	+7	+7	+7
1 000	8	-0	-7	-8
1 001	9	-1	-6	-7
1 010	10	-2	-5	-6
1 011	11	-3	-4	-5
1 100	12	-4	-3	-4
1 101	13	-5	-2	-3
1 110	14	-6	-1	-2
1 111	15	-7	-0	-1

**Figura 5.1.** Tabla resumen de las distintas formas de representación conjunta de números positivos y negativos. Pasamos de binario al número equivalente en decimal.

En la tabla de la *figura 5.1* se representan las 16 configuraciones binarias posibles para palabras de cuatro bits. En la parte derecha aparecen los distintos valores decimales equivalentes

dependiendo del modo de representación usado para interpretar el significado de esas 16 configuraciones binarias. Así, en la columna "*binario puro*" aparece el valor decimal equivalente cuando la palabra de cuatro bits se interpreta en la codificación binaria usual, sin admitir números negativos. En las otras tres columnas, sólo nos quedan tres bits para la magnitud, por lo que representamos del 0 al 7 y del -0 al -7 (en *S-M* y *C-1*) ó del 0 al 8 y del -1 al -7 (en *C-2*). Por ejemplo, la configuración 1001, si nos dicen que está en binario puro decimos que su equivalente decimal es 9. En cambio en *S-M* su equivalente decimal es -1, en *C-1* es -6 y en *C-2* es -7.

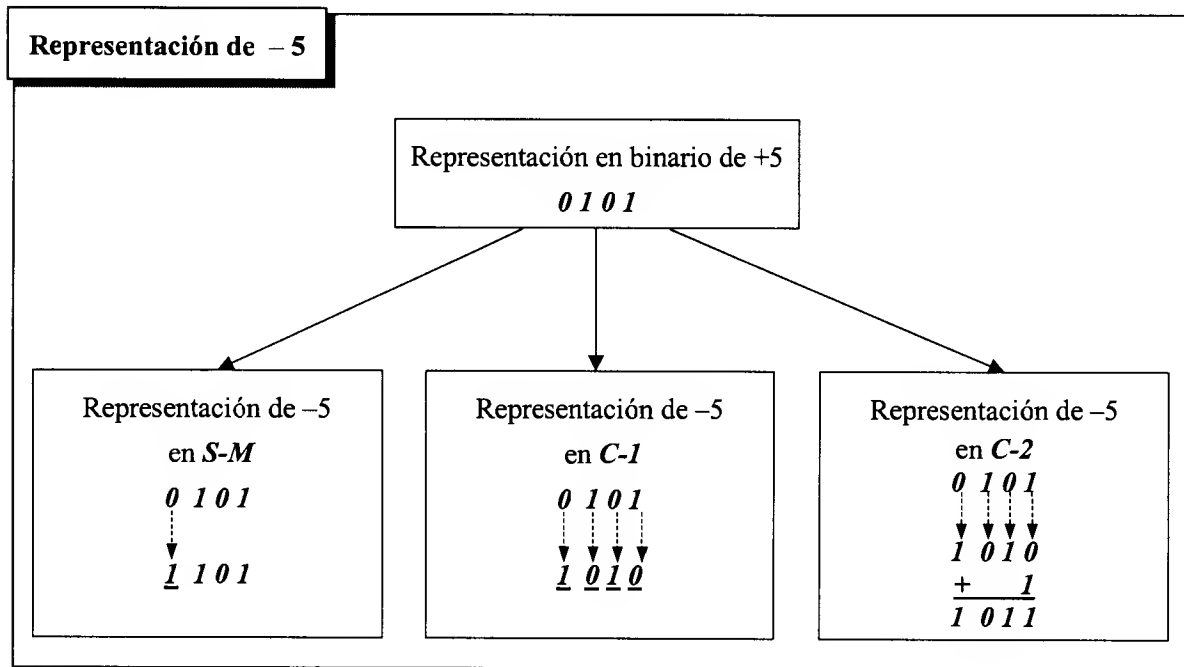
Decimal	Signo y Magnitud (S-M)	Complemento a 1 (C-1)	Complemento a la base 2 (C-2)
7	0111	0111	0111
6	0110	0110	0110
5	0101	0101	0101
4	0100	0100	0100
3	0011	0011	0011
2	0010	0010	0010
1	0001	0001	0001
0	0000 ó 1000	0000 ó 1111	0000
-1	1001	1110	1111
-2	1010	1101	1110
-3	1011	1100	1101
-4	1100	1011	1100
-5	1101	1010	1011
-6	1110	1001	1010
-7	1111	1000	1001
-8	(*)	(*)	1000

(\*) Hace falta un bit más para representarlo

**Figura 5.2.** Obtención de las tres representaciones (*S-M*, *C-1* y *C-2*) para un conjunto de números en representación decimal.

La tabla de la *figura 5.2* representa el proceso inverso al de la *figura 5.1*. Ahora partimos de los números en decimal y obtenemos las 16 configuraciones binarias equivalentes en cada uno de los tres convenios de representación (*S-M*, *C-1* y *C-2*). Obsérvese que desde +7 hasta -7 no hay problemas, pero para representar -8 ya nos hace falta un bit más, salvo en *C-2*.

Finalmente, en la *figura 5.3* se ilustra de forma resumida el proceso de obtención de la representación de un número negativo (por ejemplo, -5) en cada uno de los tres modos de representación que ya conocemos. En todos los casos partimos de la representación del número positivo correspondiente (+5, por ejemplo). Después realizamos las transformaciones individuales correspondientes para cada representación. Así, en la de M-S cambiamos el primer bit a la izquierda (bit de signo). En C-1 se complementan todos los bits y en C-2, se complementan también todos los bits y se suma 1 al resultado.



**Figura 5.3.** Ilustración del proceso de obtención de las tres representaciones de -5 (*S-M*, *C-1* y *C-2*).

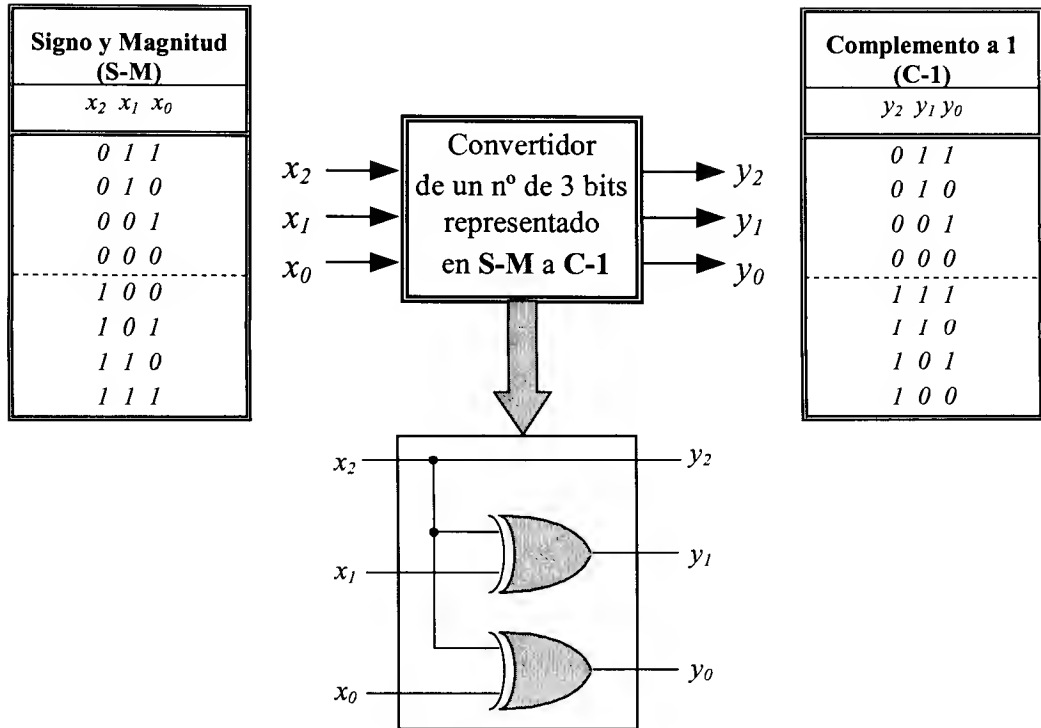
Estos procedimientos de representación terminan en configuraciones binarias fáciles de intercambiar mediante circuitos lógicos sencillos. Por ejemplo, si queremos pasar de forma automática de la representación *S-M* a *C-1* para palabras de tres bits, por simple inspección de las tablas de conversión (*figura 5.4.*), las expresiones de  $y_2$ ,  $y_1$  e  $y_0$  (de *C-1*) en función de la suma de términos mínimos de  $x_2$ ,  $x_1$  y  $x_0$  (de *S-M*) son:

$$y_2 = x_2$$

$$y_1 = \bar{x}_2 x_1 x_0 + \bar{x}_2 x_1 \bar{x}_0 + x_2 \bar{x}_1 \bar{x}_0 + x_2 \bar{x}_1 x_0 = x_2 \oplus x_1$$

[5.1]

$$y_0 = \bar{x}_2 x_1 x_0 + \bar{x}_2 \bar{x}_1 x_0 + x_2 \bar{x}_1 \bar{x}_0 + x_2 x_1 \bar{x}_0 = x_2 \oplus x_0$$



**Figura 5.4.** Circuito convertidor de S-M a C-1.

Podemos usar cualquiera de estas representaciones. Lo importante es que mantengamos la consistencia e interpretemos adecuadamente los resultados. El problema fundamental es el del “rebose” (*overflow*) que se produce cuando el resultado de una operación genera un número que no cabe en la representación para un cierto tamaño de palabra. Es decir, hacen falta más bits para poder representar el resultado de la operación de suma. Por ejemplo,  $(-7) + (-7) = -14$ .

La suma de números en C-2 se obtiene a partir de una misma operación de suma sobre todos los bits de los operandos. Sólo puede haber rebose cuando ambos operandos son positivos (bits de signo = 0) y la suma es negativa (signo = 1), o bien cuando ambos operandos son

negativos (bit de signo = 1) y la suma es positiva (signo = 0). Estas situaciones se detectan comprobando el signo del resultado. Si este signo es correcto, no hay rebose.

## 5.2. SUMADORES Y RESTADORES.

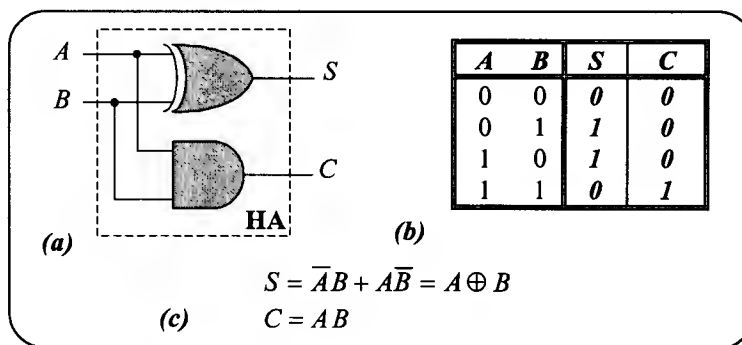
Hemos visto las distintas formas de representar números positivos y negativos con palabras binarias de  $n$  bits. Ahora tenemos que ver cómo pueden realizarse las distintas operaciones aritméticas con circuitos combinacionales. Las cuatro operaciones aritméticas básicas son suma, resta, multiplicación y división, pero si no estamos preocupados por la velocidad, las operaciones básicas son las de suma y resta porque con ellas podemos sintetizar todas las demás.

### 5.2.1. Semisumadores

Empecemos viendo la síntesis de los circuitos “semi-sumadores” (HA), llamados así porque realizan sólo la suma de dos palabras de un bit cada una ( $A$  y  $B$ ) dando como salida su suma binaria ( $S$ ) y el arrastre ( $C$ ).

Su síntesis es inmediata a partir de las tablas de verdad de la *figura 5.5.b*. En cuanto hay un bit en alta,  $S$  es “1” y cuando están los dos,  $S$  es “0”, pero  $C$  es “1”. Es decir, el arrastre es el producto lógico y la suma el OR exclusivo,

$$C = A \cdot B \quad \text{y} \quad S = A \cdot \bar{B} + \bar{A} \cdot B = A \oplus B \quad [5.2]$$



**Figura 5.5.** Semisumador. (a) Circuito. (b) Tablas de verdad. (c) Expresiones de la suma y el arrastre.

### 5.2.2. Sumadores

Si queremos sumar palabras de  $n$  bits tendremos que proporcionar un método de encadenar varios sumadores de *un* bit. Tal como opera el “semi-sumador” sólo tiene en cuenta dos entradas,  $A$  y  $B$ , pero no el potencial acarreo procedente de la suma de los bits menos significativos anteriores. El circuito que toma en cuenta ese acarreo se llama *sumador completo* (FA) y es un circuito combinacional con tres entradas  $A$ ,  $B$  y  $C_{i-1}$  y dos salidas  $S_i$  y  $C_i$ . Veámoslo formalmente.

Dos números binarios de  $n$  bits,  $A = A_{n-1} A_{n-2} \dots A_0$  y  $B = B_{n-1} B_{n-2} \dots B_0$  se suman en base dos de la forma que se ilustra en la *figura 5.6.a*. Se empieza sumando los bits menos significativos ( $A_0 + B_0$ ) que dan lugar a una suma parcial ( $S_0$ ) y a un arrastre ( $C_1$ ), siguiendo las leyes del semisumador vistas anteriormente. Este bit  $C_1$  se arrastra ahora a la segunda columna y se suma con los segundos bits,  $A_1$  y  $B_1$ , dando lugar a una nueva suma parcial  $S_1$  y a un nuevo arrastre  $C_2$ , que se suma a  $A_2$  y  $B_2$  y así sucesivamente hasta llegar a la columna de los bits más significativos ( $A_{n-1}$ ,  $B_{n-1}$ ) que junto con el penúltimo arrastre,  $C_{n-1}$ , producen el último arrastre ( $C_n$ ) de forma que el resultado es

$$S = A + B = C_n S_{n-1} S_{n-2} \dots S_1 S_0 \quad [5.3]$$

Observando el procedimiento usado vemos que para su síntesis necesitaríamos disponer de un módulo que sumara tres bits ( $A_i$ ,  $B_i$ ;  $C_i$ ) y que produzca una suma local y un arrastre ( $S_i$ ,  $C_{i+1}$ ). A este circuito es al que hemos llamado “*sumador completo*” y satisface la tabla de verdad de la *figura 5.6.b*.

Las funciones correspondientes se obtienen por simplificación de su expresión en términos mínimos que da lugar a la síntesis mediante dos semisumadores ilustrada en la *figura 5.6.c*.

$$S_i = \overline{C_i} (\overline{A_i} B_i + A_i \overline{B_i}) + C_i (\overline{A_i} \overline{B_i} + A_i B_i) \quad [5.4]$$

$$\text{si llamamos } (\overline{A_i} B_i + A_i \overline{B_i}) = A_i \oplus B_i = D_i \quad [5.5]$$

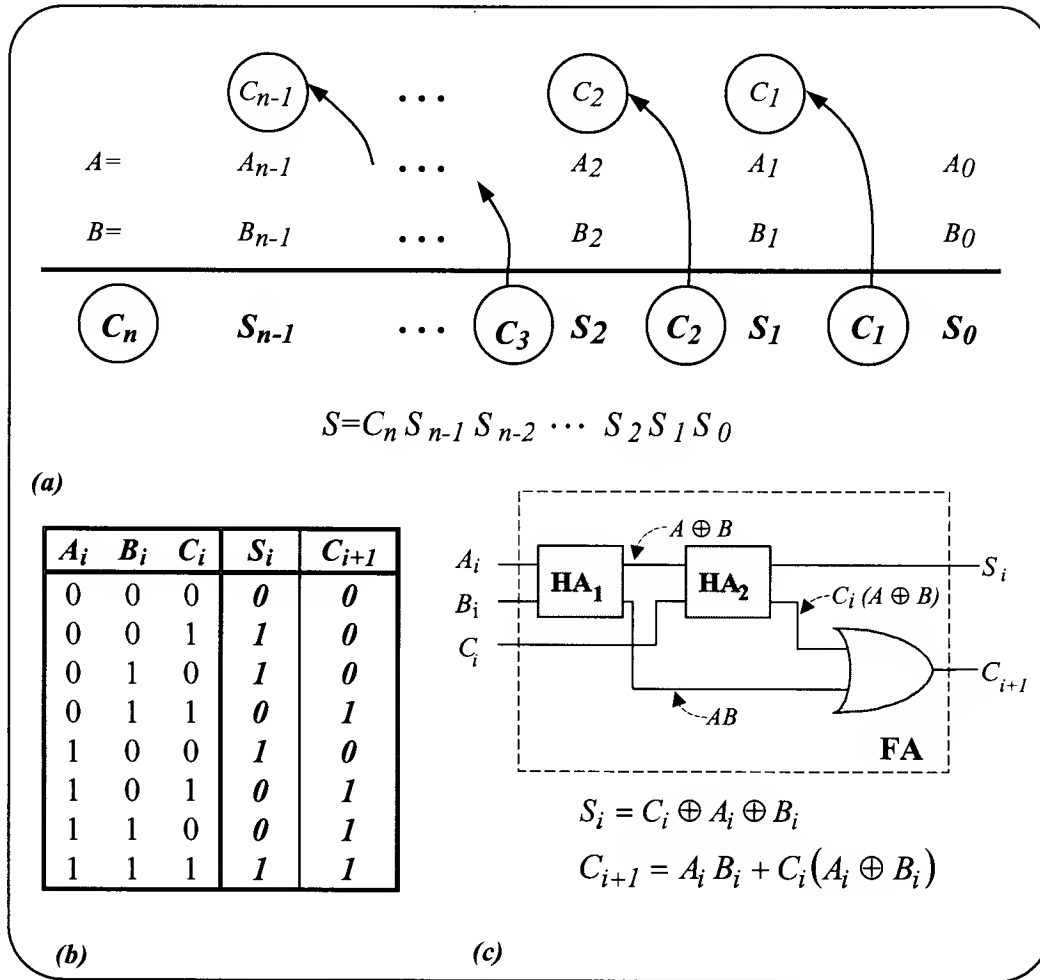
$$(\overline{A_i} \overline{B_i} + A_i B_i) = \overline{A_i \oplus B_i} = \overline{D_i} \quad [5.6]$$

obtenemos de forma más compacta:

$$S_i = \bar{C}_i D_i + C_i \bar{D}_i = C_i \oplus D_i = C_i \oplus A_i \oplus B_i \quad [5.7]$$

Análogamente, para el arrastre obtenemos:

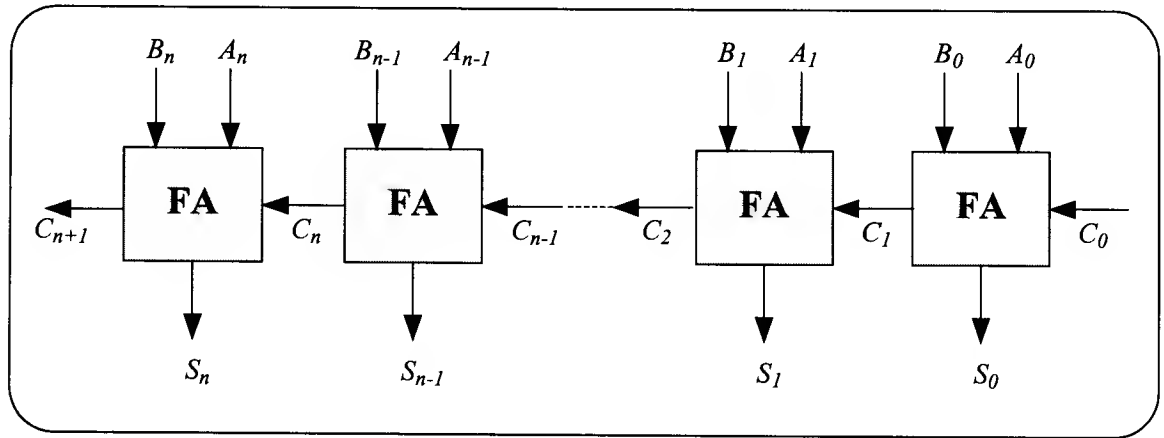
$$\begin{aligned} C_{i+1} &= \bar{C}_i A_i B_i + C_i \bar{A}_i B_i + C_i A_i \bar{B}_i + C_i A_i B_i = \\ &= \bar{C}_i A_i B_i + C_i (A_i \oplus B_i) + C_i A_i B_i = A_i B_i + C_i (A_i \oplus B_i) \end{aligned} \quad [5.8]$$



**Figura 5.6.** Sumador completo. (a) Algoritmo de suma de dos números de  $n$  bits. (b) Síntesis a partir de semisumadores. (c) Tabla de verdad del sumador completo.



Con el sumador completo hemos resuelto el problema de la suma para *dos* bits más el acarreo. Sin embargo, la necesidad real es sumar dos números de  $n$  bits, de acuerdo con el procedimiento ilustrado en la *figura 5.6*. Esto se puede hacer usando varios sumadores completos conectados en paralelo (*figura 5.7*) o usando un único sumador completo en un circuito que incluya un retardo y un lazo de realimentación (conexión serie de la *figura 5.11*), para almacenar el acarreo generado en cada paso.



**Figura 5.7.** Extensión del sumador completo de la *figura 5.6.c* a un sumador paralelo para palabras de 4 bits.

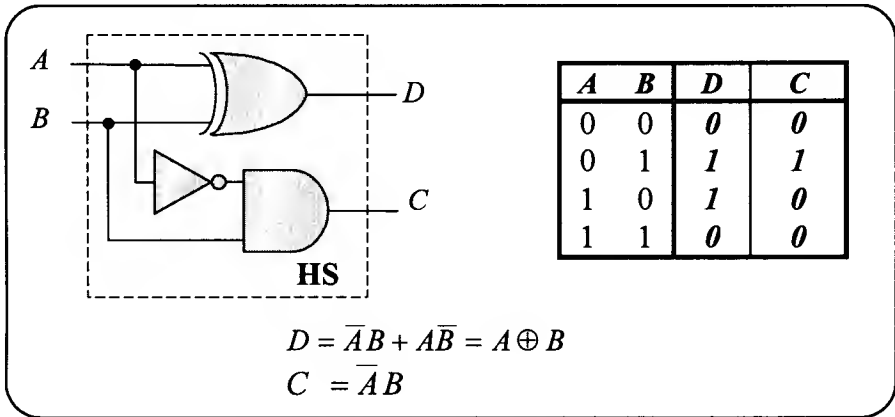
En la solución paralelo se utilizan  $n$  sumadores completos y todos los bits de los números binarios a sumar entran simultáneamente al circuito. El acarreo de salida de cada sumador de bit local se conecta (en el caso más sencillo) a la entrada de acarreo de la etapa siguiente. Esta forma de operación exige que todos los bits de ambos números se encuentren disponibles en el mismo instante y permanezcan allí hasta que se hayan producido todos los acarros (venciendo el retardo de propagación de la familia lógica correspondiente) y la suma correcta se encuentre disponible y estable en los terminales de salida.

Es fácil comprender las limitaciones que introduce este retardo en la velocidad del sumador, ya que hay que esperar hasta que el acarreo atraviese los  $n$  módulos desde la entrada (bit menos significativo) hasta la salida. La solución de este problema ha dado lugar a nuevos diseños de sumadores rápidos, que resuelven de forma simultánea la contribución a la suma de todos los acarros locales. Veremos alguna de estas soluciones tras comentar el funcionamiento

de los sumadores serie y los semirrestadores y restadores, con lo que completamos la descripción de los circuitos aritméticos en su primer nivel de complejidad.

### 5.2.3. Semirrestadores

Razonando de forma análoga a como hicimos con los sumadores podemos ahora proponer un circuito semirrestador para palabras de un bit. Si llamamos  $A$  al minuendo y  $B$  al sustraendo, la *figura 5.8* incluye de forma resumida la tabla de verdad, el circuito y las ecuaciones lógicas correspondientes al bit de diferencia ( $D$ ) y al acarrea ( $C$ ).



**Figura 5.8.** Semirrestador

### 5.2.4. Restadores Completos

Si tenemos en cuenta no sólo el minuendo ( $A_i$ ) y el sustraendo ( $B_i$ ) sino el arrastre potencial de una resta anterior ( $C_i$ ) podemos ahora generar la diferencia ( $D_i$ ) y el nuevo arrastre para la etapa siguiente ( $C_{i+1}$ ), dando lugar a un circuito restador completo, como el que se ilustra en la *figura 5.9*, donde hemos incluido también la tabla de verdad y las ecuaciones.

Si ahora encadenamos los acarrees de  $n$  restadores completos de palabras de un bit obtenemos un restador paralelo para palabras de  $n$  bits. La *figura 5.10*, muestra el circuito resultante.

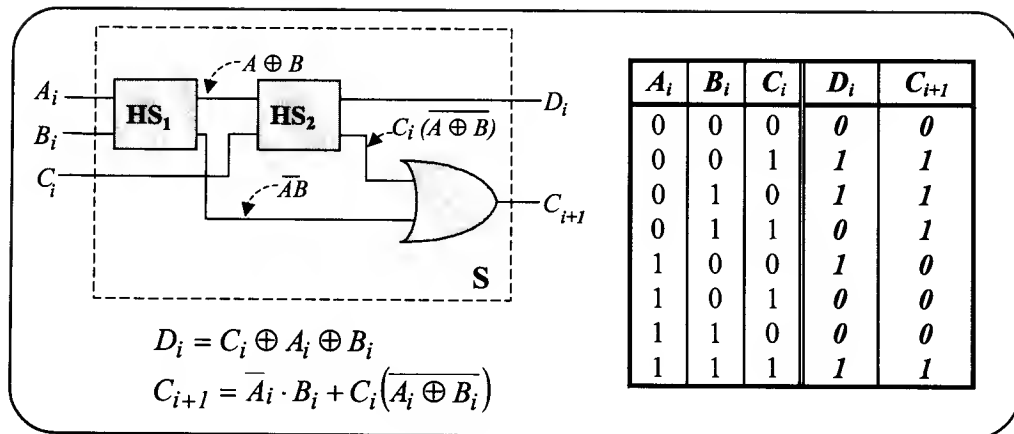


Figura 5.9. Restador.

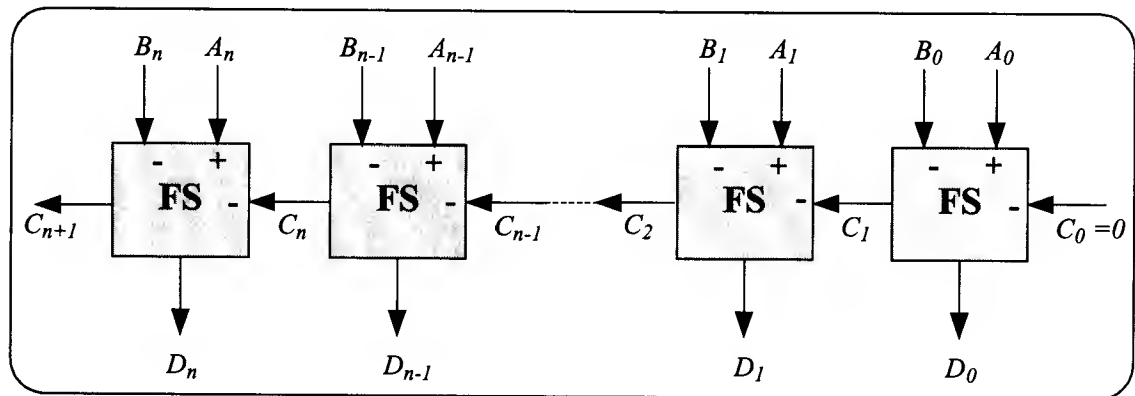
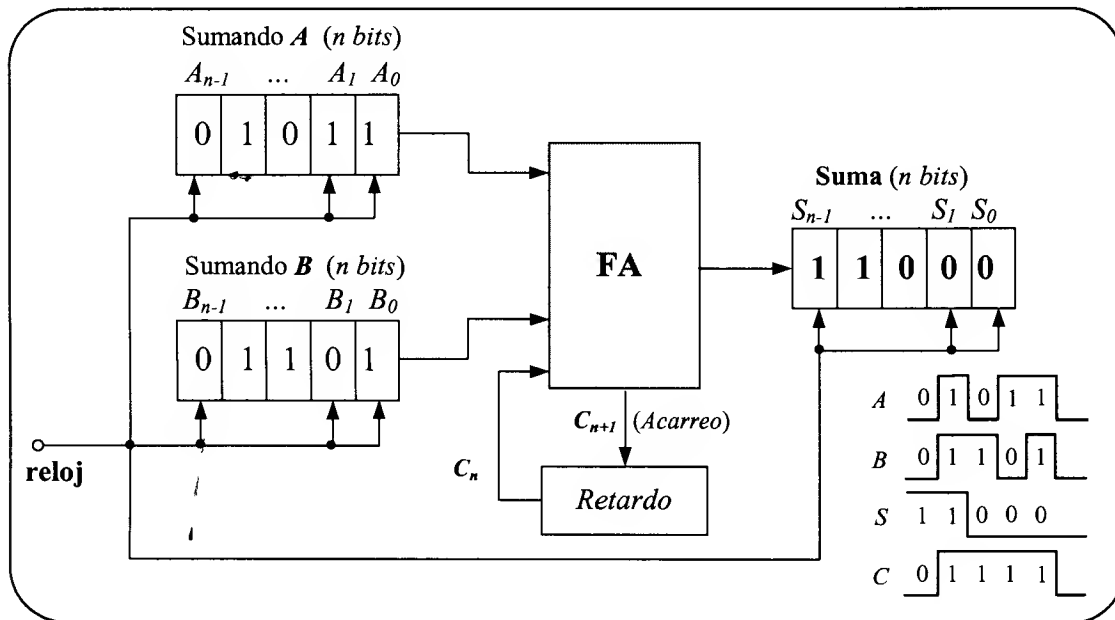


Figura 5.10. Restador paralelo

### 5.2.5. Sumador Serie

Terminamos la primera fase en la complejidad de los circuitos sumadores viendo cómo podría realizarse un sumador serie en el que los distintos bits de las dos palabras a sumar,  $A$  y  $B$ , entran al sumador como trenes de impulsos ( $A_{n-1}, A_{n-2}, \dots, A_1, A_0$ ) y ( $B_{n-1}, B_{n-2}, \dots, B_1, B_0$ ) de forma que hay un reloj que controla la entrada síncrona de estos dos trenes de impulsos a un sumador completo. Además, los acarrees que va generando el sumador se almacenan

transitoriamente en un registro que se encarga de retardarlos para presentarlos a la entrada cuando llegan los siguientes bits. La *figura 5.11* muestra el sumador serie y las formas de onda para un ejemplo concreto de suma. Comparando este circuito con los anteriores que sumaban en paralelo se observa que este es más lento, porque allí todos los bits se suman simultáneamente, salvo el problema de los arrastres, y aquí hay un retardo por bit y los números que se suman han debido ser almacenados previamente en registros de desplazamiento.



**Figura. 5.11.** Esquema cualitativo de un sumador serie para palabras de  $n$  bits.

### 5.2.6. Sumador Paralelo con Acarreo Adelantado

Los sumadores con acarreo enlazado que hemos visto previamente tienen el problema de la baja velocidad debido a la forma de propagar el acarreo. Hay que esperar a que llegue el acarreo del penúltimo bit al sumador completo del último bit para dar por finalizada la suma. Para resolver ese problema se usa la técnica de adelantar el acarreo (look-ahead-carry), base de las soluciones integradas en MSI tipo SN7483. Consideremos de nuevo el circuito del sumador completo de la *figura 5.6.c* y las señales intermedias  $P$  y  $G$  que salen del primer semisumador.

$$P_i = A_i \oplus B_i \quad y \quad G_i = A_i \cdot B_i \quad [5.9]$$

Las salidas del sumador se pueden expresar en términos de  $P$  y  $G$  como

$$S_i = P_i \oplus C_i \quad y \quad C_i = G_i + P_i C_i \quad [5.10]$$

La señal interna,  $G_i$ , se denomina "*generación de acarreo*" y produce un estado de alta cuando  $A_i = B_i = "1"$ . La señal  $P_i$  se denomina de "*propagación del acarreo*" y es la que se compone con el acarreo de entrada ( $C_i$ ) para producir el de salida ( $C_{i+1}$ ).

Escribamos ahora las funciones para la salida de arrastre de cada etapa en función de las señales  $P$  y  $G$ .

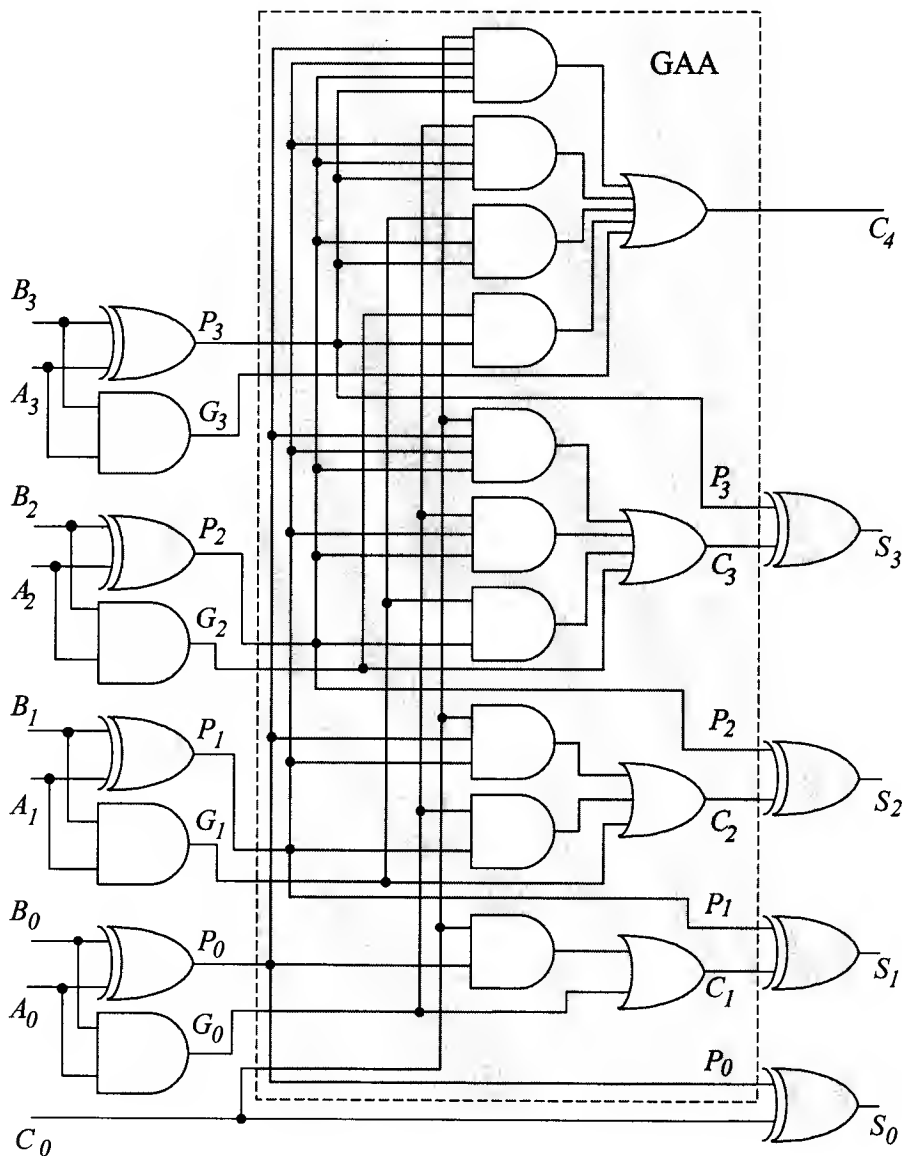
$$\begin{aligned} C_1 &= G_0 + P_0 C_0 \\ C_2 &= G_1 + P_1 C_1 = G_1 + P_1 (G_0 + P_0 C_0) = G_1 + P_1 G_0 + P_1 P_0 C_0 \\ C_3 &= G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0 \\ C_4 &= G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0 \end{aligned} \quad [5.11]$$

Como estas salidas de arrastre aparecen en forma de suma de productos, se pueden sintetizar mediante dos niveles y por consiguiente podemos tener sus valores disponibles con sólo dos retardos. El circuito de la *figura 5.12* muestra la síntesis de estas señales de "acarreo adelantado" y su uso en la construcción de un sumador paralelo de cuatro bits añadiendo al acarreo adelantado la parte de circuito correspondiente a los semisumadores de los 4 bits. Obsérvese que los cuatro bits de salida,  $S_0$  a  $S_3$ , sufren el mismo retardo de propagación, el correspondiente a los cuatro niveles de puertas lógicas usados:

- a) Los semisumadores que generan las señales  $P$  y  $G$ .
- b) Las puertas AND que generan los términos mínimos del acarreo adelantado.
- c) Las puertas OR.
- d) Las XOR de salida

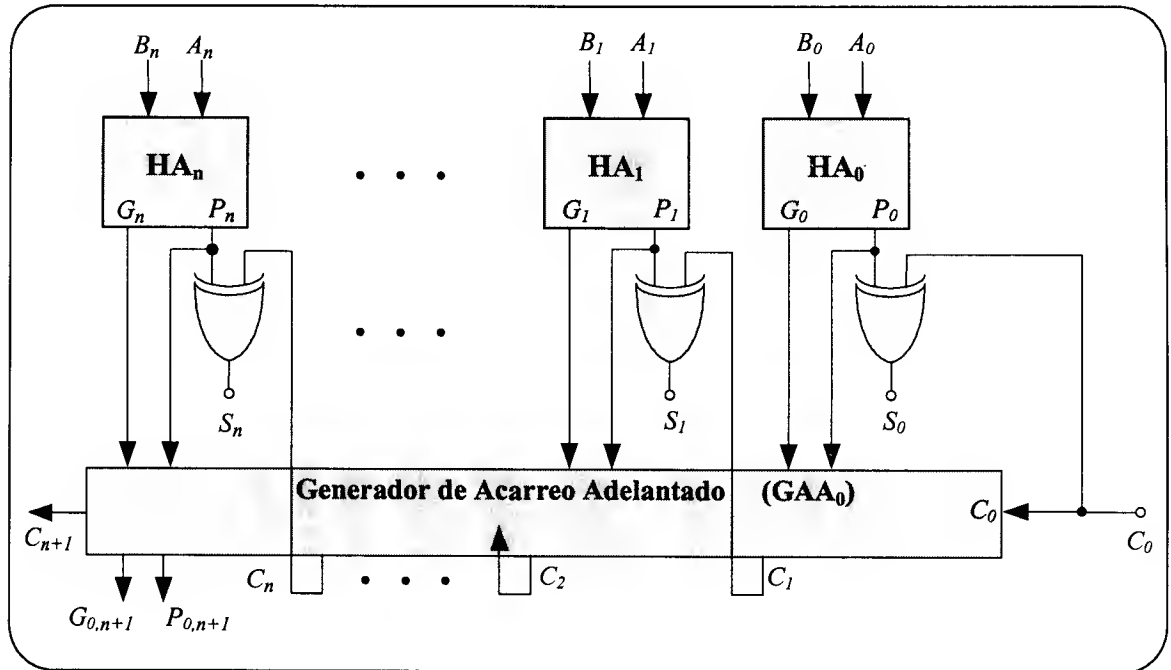
El circuito de la *figura 5.12* conviene estudiarlo en dos etapas. Primero centrándose sólo en las puertas incluidas en el rectángulo punteado que son las encargadas de la síntesis del acarreo adelantado. Después, el circuito completo, incluyendo los semisumadores de entrada y

los circuitos XOR de salida. Obsérvese que el circuito posee como entradas ( $A_0...A_3$ ), ( $B_0...B_3$ ) y  $C_0$  y produce como salidas ( $S_0...S_3$ ) y  $C_4$ . Este último bit de arrastre permite la conexión modular para ampliar la capacidad de suma usando nuevos circuitos. La *figura 5.13* repite el esquema del sumador de forma más compacta, sin especificar los circuitos encargados de adelantar el acarreo.



**Figura 5.12.** Sumador paralelo de 4 bits con lógica de acarreo adelantado.

El aumento de velocidad conseguido con el acarreo adelantado será tanto mayor, cuanto mayor sea la longitud de las palabras a sumar. Sin embargo las restricciones de conectividad (fan-in y fan-out) de las distintas familias lógicas ponen un límite a la máxima longitud de las palabras a sumar. Obsérvese por ejemplo que  $G_1$  se usa en el cálculo de  $C_2$ ,  $C_3$  y  $C_4$ . Una solución a este problema es sumar por bloques de forma que dentro de cada bloque se opera con acarreo adelantado y entre bloques se usa el acarreo enlazado que estudiamos anteriormente.



**Figura 5.13.** Versión compacta del circuito de sumador paralelo con acarreo adelantado de la figura 5.12.

Hay otras formas más avanzadas de obtener sumadores de alta velocidad que quedan fuera del alcance de este texto.

### 5.3. SUMADORES EN COMPLEMENTO A 1: GESTIÓN DEL PROBLEMA DEL REBOSE

Hemos visto los circuitos sumadores en binario sin preocuparnos del problema del rebose.

Vamos a estudiar aquí las modificaciones necesarias en un circuito sumador que acepta números positivos y negativos en representación por complemento a 1. Y lo vamos a hacer sobre palabras de sólo dos bits para poder seguir el procedimiento en extenso, usando tablas de verdad. Después podremos generalizar el algoritmo, viendo cuándo hay que añadir un *uno* al resultado y cuándo se produce error de rebose.

Recordar del apartado inicial de este tema que:

- I. Para obtener los números negativos en C-1 hay que cambiar los “0” por “1” y los “1” por “0”.
- II. Para sumar números positivos y negativos en C-1, se suma en binario y se desprecia el bit de acarreo cuando éste es cero. Cuando es uno se desprecia también, pero en este caso se le suma un uno al resultado previo.

En este caso, como queremos encontrar un circuito que sume palabras de sólo dos bits,  $(A_1, A_0)$  y  $(B_1, B_0)$ , sólo tenemos 2 bits para representar los números positivos y negativos y el bit más significativo es el bit de signo. Por consiguiente, realmente sólo nos queda 1 bit para representar las magnitudes. Así los números que podemos representar en C-1 (con 2 bits) son:

C-1	00	01	10	11
Decimal	+0	+1	-1	-0

Veamos ahora algunos ejemplos antes de abordar la síntesis del circuito:

- Sin problemas ni necesidad de sumar “1” al resultado:

$$\begin{array}{rcl}
 00 & (+0) & \\
 + 01 & (+1) & \\
 \hline
 01 & \rightarrow & (+1)
 \end{array}
 \qquad
 \begin{array}{rcl}
 01 & (+1) & \\
 + 10 & (-1) & \\
 \hline
 11 & \rightarrow & (-0)
 \end{array}$$

- Con necesidad de sumar “1” al resultado:

$$\begin{array}{rcl}
 10 & (-1) & \\
 + 11 & (-0) & \\
 \hline
 1\ 01 & \rightarrow & (+1)
 \end{array}$$



Obsérvese que si nos quedamos con los dos últimos bits (01), el resultado sería erróneo ya que 01 corresponde a +1 en C-1. Por consiguiente hay que sumarle “1” al resultado Así,

$$\begin{array}{r}
 10 \quad (-1) \\
 + \quad 11 \quad (-0) \\
 \hline
 101 \\
 \downarrow +1 \\
 \hline
 10 \rightarrow (-1)
 \end{array}$$

➤ *Con problemas de rebose:*

$$\begin{array}{r}
 01 \quad (+1) \\
 + \quad 01 \quad (+1) \\
 \hline
 010 \rightarrow (-1)
 \end{array}$$

Aquí hay error de rebose porque (+1)+(+1) es +2 y nos sale -1 (lógicamente en C-1). Esto es debido a que nos hace falta un bit más. Entonces, con tres bits, (010, +2 en decimal) sería correcta la suma. El bit de signo sería el 0 (+) y el de magnitud los otros dos, 10 (+2).

Así pues, al diseñar el circuito sumador en C-1 tenemos que detectar las configuraciones que generan este error de rebose. Para ello vamos a construir la tabla de verdad completa (*figura 5.14*).

Obsérvese que el rebose sólo se puede producir cuando *los dos sumandos son del mismo signo* (ambos positivos o ambos negativos) pues cuando son de signo distinto el resultado siempre será menor que uno de los operandos y podrá representarse usando el mismo número de bits que hemos usado para los operandos. Para saber si se produce o no el rebose hay que comprobar el *signo de la suma*. Si es correcto no hay rebose. Si es incorrecto, si.

Resumen de las distintas funciones que debe realizar el circuito.

- I. *Sumar en binario puro todos los bits sin distinguir entre bit de signo y bits de magnitud.*
- II. *Dar por válido el resultado si el acarreo es cero y no hay rebose.*
- III. *Si el acarreo es 1 y no hay rebose, sumar un 1 al resultado de I y dar por válido este nuevo resultado.*

IV. Si hay rebose (porque el signo de la operación no es correcto) dar señal de error.

La ecuación del rebose es:

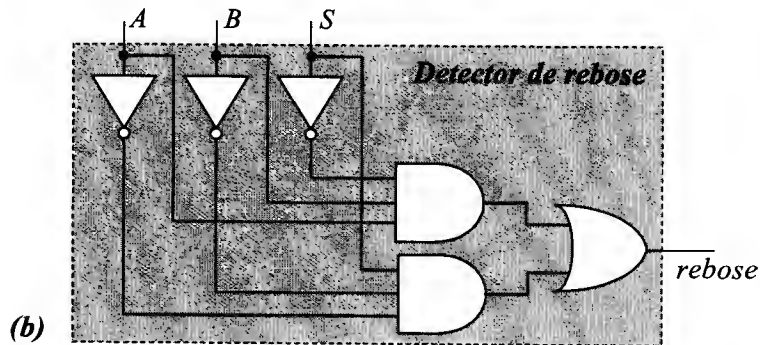
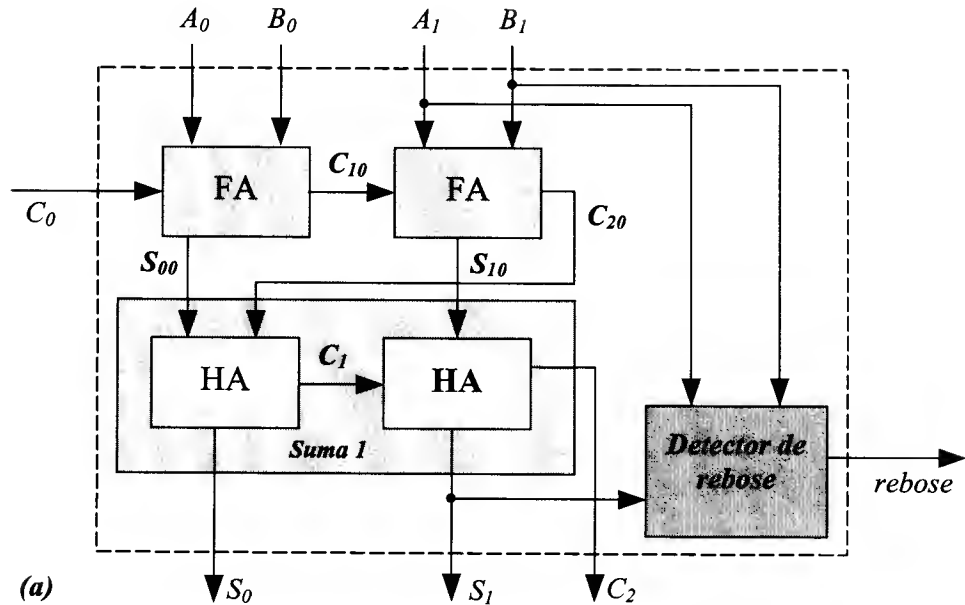
$$rebose = \overline{A_1} \overline{B_1} S_1 + A_1 B_1 \overline{S_1}$$

[5.12]

n° decimal	B <sub>1</sub>	B <sub>0</sub>	n° decimal	A <sub>1</sub>	A <sub>0</sub>	C <sub>2</sub>	S <sub>1</sub>	C <sub>1</sub>	S <sub>0</sub>	n° decimal
(+0)	0	0	(+0)	0	0	0	0	0	0	(+0)
	0	0	(+1)	0	1	0	0	0	1	(+1)
	0	0	(-1)	1	0	0	1	0	0	(-1)
	0	0	(-0)	1	1	0	1	0	1	(-0)
(+1)	0	1	(+0)	0	0	0	0	0	1	(+1)
	0	1	(+1)	0	1	0	1	1	0	(-1) rebose
	0	1	(-1)	1	0	0	1	0	1	(-0)
	0	1	(-0)	1	1	1	0	1	0	
						+			1	
						0		1		(+1)
(-1)	1	0	(+0)	0	0	0	1	0	0	(-1)
	1	0	(+1)	0	1	0	1	0	1	(-0)
	1	0	(-1)	1	0	1	0	0	0	
						+			1	
						0		1		(+1) rebose
	1	0	(-0)	1	1	1	0	0	1	
						+			1	
						1		0		(-1)
(-0)	1	1	(+0)	0	0	0	1	0	1	(-0)
	1	1	(+1)	0	1	1	0	1	0	
						+			1	
						0		1		(1)
	1	1	(-1)	1	0	1	0	0	1	
						+		1	1	
						1		0		(-1)
	1	1	(-0)	1	1	1	1	1	0	
						+			1	
						1		1		(-0)

Figura 5.14. Descripción en extenso de las distintas situaciones posibles en la suma por C-1 de dos palabras de 2 bits.

El circuito completo es el de la *figura 5.15*.



**Figura 5.15.** Suma por *C-1*. (a) Esquema del circuito a partir de sumadores y semisumadores. (b) Circuito detector del rebose.

## 5.4. COMPARADORES

La operación lógica de comparación es esencial en todo el cálculo digital. Está en la base

de la ramificación condicional y aparece en funciones de búsqueda e identificación selectiva de áreas de datos, frecuente en editores, ensambladores y compiladores.

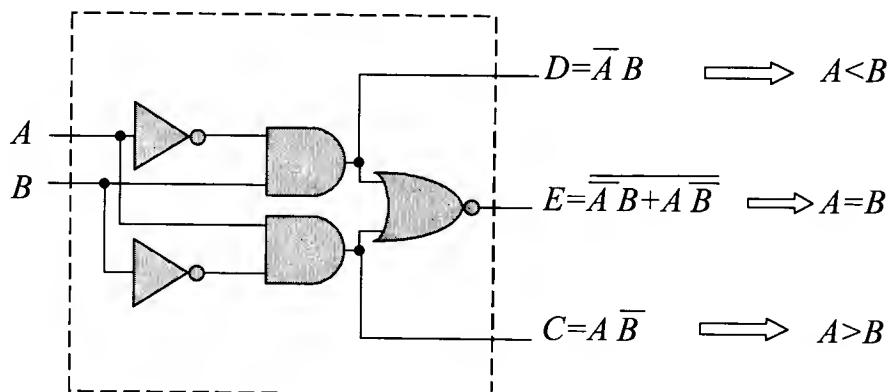
Un comparador de dos palabras de  $n$  bits ( $A$  y  $B$ ) es un circuito que determina cual de estas palabras es mayor, cuál es menor y cuándo son iguales. Tiene que producir entonces tres salidas:

$$A > B, \quad A = B, \quad A < B$$

La base para la síntesis de comparadores es el circuito "coincidencia" que sólo está activo cuando ambos bits son iguales ( $A = B = "0"$ ,  $A = B = "1"$ ). Es el complementario del OR-exclusivo ( $A \oplus B = 1$ ,  $\overline{A \oplus B} = 0$ ).

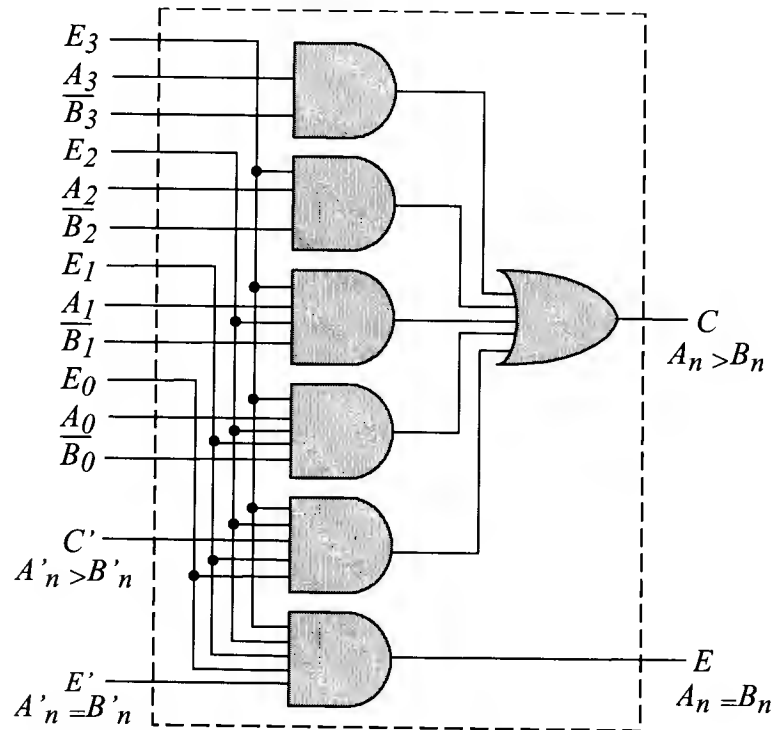
$$E = \overline{A \oplus B} = \begin{cases} 1 & \text{si } A = B \\ 0 & \text{si } A \neq B \end{cases} \quad [5.13]$$

La condición  $A > B$  se detecta a partir de la función lógica  $C = A \cdot \overline{B} = 1$ , ya que si es  $A$  mayor que  $B$ , entonces  $A=1$  y  $B=0$ , y por consiguiente,  $C=1$ . Por otro lado si  $A=B$  ó  $A$  es menor que  $B$  ( $A=0$ ,  $B=1$ ), entonces,  $C=0$ . De forma análoga, la condición  $A < B$ , se obtiene a partir de la función  $D = \overline{A} \cdot B = 1$ . La figura 5.16 muestra el circuito comparador correspondiente a un bit que genera las salidas  $E$ ,  $C$  y  $D$ .



**Figura 5.16.** Circuito comparador de dos "palabras" de un bit.

Para ampliar esta comparación a palabras de  $n$  bits necesitamos pasar las condiciones de igualdad o desigualdad de un bit a las condiciones de igualdad o desigualdad de toda la palabra que, evidentemente, son funciones lógicas de los  $n$  bits. Así, por ejemplo, el circuito de la figura 5.17 muestra estas funciones para un comparador de palabras de 4 bits.



**Figura 5.17.** Extensión del circuito comparador a palabras de cuatro bits.

Si  $A = B$  tendrá que ser:  $A_3=B_3$ ,  $A_2=B_2$ ,  $A_1=B_1$  y  $A_0=B_0$  y, por consiguiente, la condición de igualdad será:

$$E = E_3 \cdot E_2 \cdot E_1 \cdot E_0 = 1$$

[5.14]

Veamos ahora las situaciones de desigualdad. Dentro de la desigualdad ( $\bar{E} = 1$ ) tenemos dos situaciones. La primera es que  $A$  sea mayor que  $B$  ( $A > B$ ) y la segunda, que se obtiene por exclusión es que  $A$  sea menor que  $B$  ( $A < B$ ).

Para que se cumpla la condición  $A > B$  pueden ocurrir dos de las siguientes situaciones:

$$\begin{aligned}
 & A_3 > B_3 \Rightarrow A_3 \bar{B}_3 \\
 \text{ó} & A_3 = B_3 \text{ y } A_2 > B_2 \Rightarrow E_3 A_2 \bar{B}_2 \\
 \text{ó} & A_3 = B_3 \text{ y } A_2 = B_2 \text{ y } A_1 > B_1 \Rightarrow E_3 E_2 A_1 \bar{B}_1 \\
 \text{ó} & A_3 = B_3 \text{ y } A_2 = B_2 \text{ y } A_1 = B_1 \text{ y } A_0 > B_0 \Rightarrow E_3 E_2 E_1 A_0 \bar{B}_0
 \end{aligned}
 \tag{5.15}$$

Finalmente, por exclusión, si  $\bar{E} = 1$  y  $C(A > B) = 0$ , se tiene que cumplir que:  $A < B$ .

Existen soluciones integradas MSI a los circuitos comparadores que van desde la SN7485 (comparador de cuatro bits que se puede conectar en cascada) hasta circuitos tipo SN74AS866A, capaz de comparar aritmética o lógicamente palabras de 8 bits (en representación binaria o en complemento a dos), con posibilidad de expansión y salida en colector abierto. La *figura 5.18* muestra el símbolo.

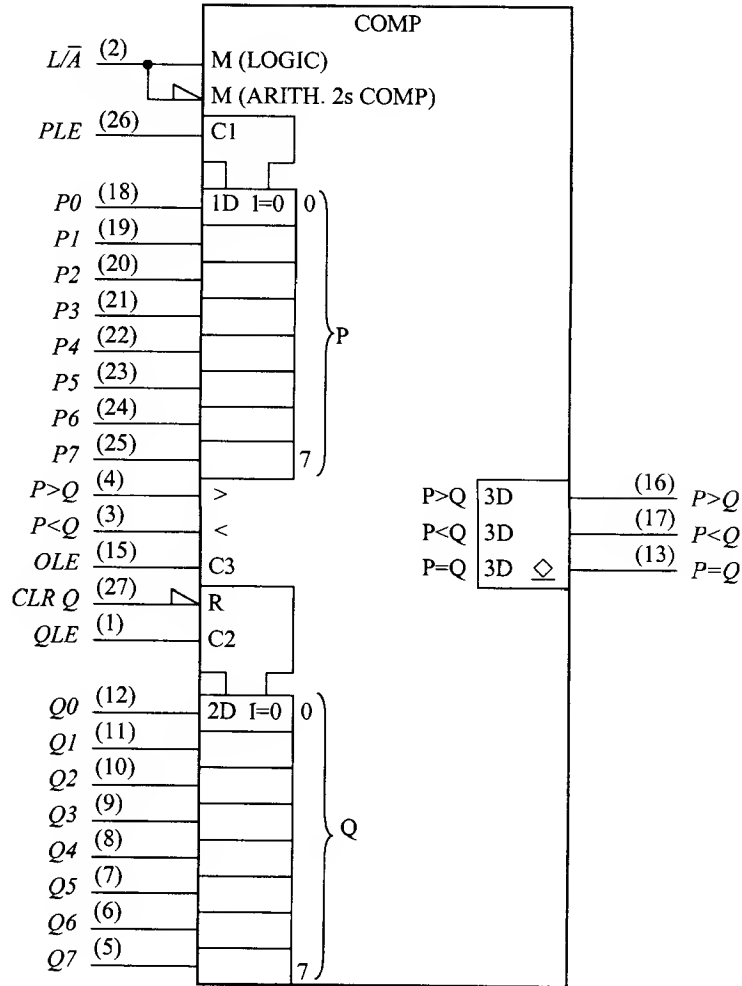
El símbolo lógico que se utiliza para describir el circuito está de acuerdo con las normas ANSI/IEEE. En el apéndice B se resume esta forma de representación porque en el manejo de entornos software para la captación de esquemas, la simulación y el diseño lógico son estos símbolos los que se usan, en general, para representar a los circuitos integrados.

Volviendo ahora a la *figura 5.18* conviene resaltar que tanto las entradas,  $P$  y  $Q$ , como las salidas ( $P > Q$ ,  $P < Q$  y  $P = Q$ ) están equipadas con circuitos biestables que permiten retener en una memoria local el resultado de la comparación durante un cierto tiempo. Posee además circuitos de facilitación ("*enable*",  $PLE$ ,  $QLE$ ,  $OLE$ ) que mejoran el comportamiento en aquellas aplicaciones que exigen la conexión en cascada de varios circuitos comparadores.

Cada salida del comparador muestra su estado cuando la señal de facilitación correspondiente está en alta. El registro  $Q$  puede ponerse a cero para acelerar el proceso de comparación de palabras  $P$  con cero en aquellas aplicaciones que sólo exijan saber si una determinada palabra es igual o distinta de cero.

El circuito SN74AS866A admite una programación mínima para seleccionar el tipo de función. La *figura 5.19* muestra la tabla de selección de función incluyendo la comparación lógica y la aritmética y en cada uno de estos casos cuáles son las entradas y las salidas correspondientes. Los símbolos que aparecen en la tabla son todos de interpretación inmediata

salvo el *AG* ("Arithmetically Greater than") que corresponde al "mayor que" en el sentido aritmético del término.



**Figura 5.18.** Circuito comparador de 8 bits, SN74AS866A. Símbolo lógico de acuerdo con las normas ANSI/IEEE.

La función de comparación se particulariza a veces a soluciones programables para comparadores de identidad, como la SN74ALS528 que permite resolver de forma eficiente un problema de comparación cuando uno de los números es fijo. En otras ocasiones, la función de

comparación se extiende a función de generación y detección de paridad, como en el caso de los circuitos SN74ALS280 y 286. El primero es un generador/detector de paridad para canales con nueve líneas de datos. El segundo, 286, realiza también la función de generación/detección de paridad para palabras de 9 bits pero añade la facilidad de conexión de varios circuitos en cascada para conseguir comparar palabras de más de 9 bits. A su vez, facilita la conexión directa a los canales de datos a través de circuitos amplificadores/adaptadores de impedancias.

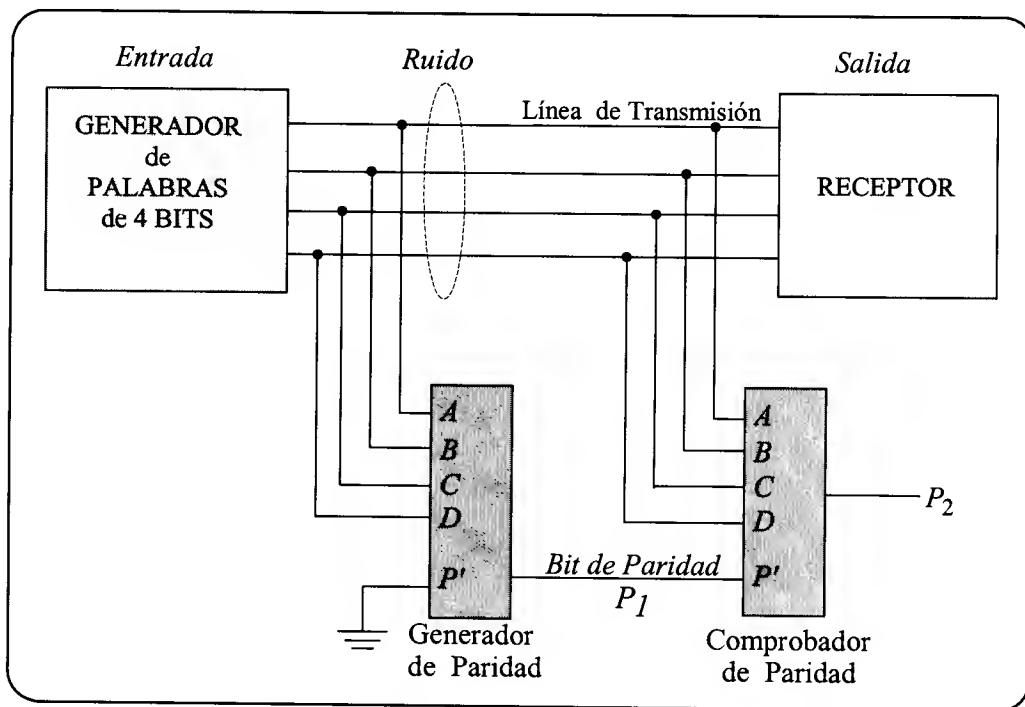
TABLA DE FUNCIÓN DEL COMPARADOR							
Comparación	L/ $\overline{A}$	Datos de Entrada  P0-P7, Q0-Q7	ENTRADAS		SALIDAS		
			P>Q	P<Q	P>Q	P<Q	P=Q
Lógica	H	P>Q	x	x	<b>H</b>	<b>L</b>	<b>L</b>
Lógica	H	P<Q	x	x	<b>L</b>	<b>H</b>	<b>L</b>
Lógica	H	P=Q	L	L	<b>L</b>	<b>L</b>	<b>H</b>
Lógica	H	P=Q	L	H	<b>L</b>	<b>H</b>	<b>L</b>
Lógica	H	P=Q	H	L	<b>H</b>	<b>L</b>	<b>L</b>
Lógica	H	P=Q	H	H	<b>H</b>	<b>H</b>	<b>L</b>
Aritmética	L	P AG Q	x	x	<b>H</b>	<b>L</b>	<b>L</b>
Aritmética	L	Q AG P	x	x	<b>L</b>	<b>H</b>	<b>L</b>
Aritmética	L	P=Q	L	L	<b>L</b>	<b>L</b>	<b>H</b>
Aritmética	L	P=Q	L	H	<b>L</b>	<b>H</b>	<b>L</b>
Aritmética	L	P=Q	H	L	<b>H</b>	<b>L</b>	<b>L</b>
Aritmética	L	P=Q	H	H	<b>H</b>	<b>H</b>	<b>L</b>

**Figura 5.19.** Tabla de selección de función en el comparador aritmético-lógico 74866. Con  $L/\overline{A}$  se selecciona el tipo de comparación (lógica o aritmética).

Los circuitos generadores/detectores de paridad se activan cuando la suma de los unos de las palabras es par o bien cuando la suma de los unos es impar. Su uso más general es la detección de errores en transmisión de datos y en funciones de bifurcación. La *figura 5.20* muestra la posición relativa de estos circuitos en un esquema general de transmisión y recepción de información.



Partimos de un generador de palabras de cuatro bits que puede ser un generador en sentido estricto o bien cualquier fuente de información como por ejemplo un registro de los que estudiaremos más adelante. La palabra se introduce en un canal y, a la vez, se manda por separado un bit de paridad. Si, como consecuencia del ruido, existiera una confusión en algún bit es muy poco probable que exista a la vez otro error en otro bit que cancele el cambio de paridad. Por consiguiente, en recepción existe un nuevo circuito comprobador de paridad que se activa siempre que la paridad de la palabra emitida no coincide con la paridad de la palabra recibida.



**Figura 5.20.** Circuitos detectores/generadores de paridad. Ilustración de su uso en transmisión.

Veamos ahora como se sintetizan los circuitos detectores de paridad. Para ello recordemos que la salida del OR exclusivo  $(A\bar{B} + \bar{A}B)$  sólo está en alta si hay un bit de entrada en alta (pero no ambos). Es decir, ante (0,1) y (1,0) da 1 (impar), pero ante (0,0) y (1,1), da 0 (par). Y este procedimiento es iterativo. Si con un XOR detectamos la no paridad de (A, B) y con otro la de (C, D), con un tercero detectaremos la correspondiente a los cuatro bits, A, B, C y D. La figura

5.21 muestra esta generalización para 4 bits. Como en Z detectamos la no paridad de A, B, C y D y ha de enlazar con la paridad de otro módulo ( $P'$ ), tendremos que añadir otra puerta XOR, para obtener en P la paridad de A, B, C, D y  $P'$ .

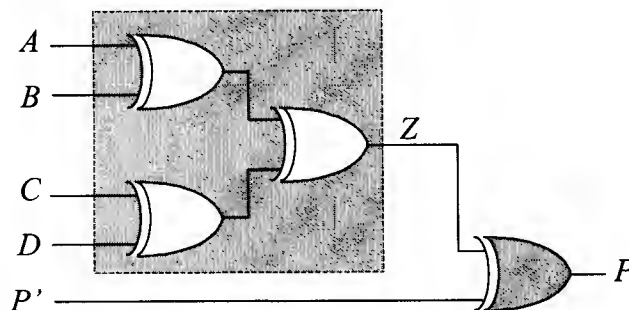
La salida en el punto Z de la *figura 5.21* es :

$$Z = (A \oplus B) \oplus (C \oplus D) \quad [5.16]$$

Así, si el número de bits que hay en alta en la palabra es (1 ó 3), entonces  $Z=1$  (*impar*). En cambio, si el número de bits en alta es (0, 2 ó 4), entonces  $Z=0$  (*par*). Por otra parte  $P'=1$  significa que la palabra anterior es par y  $P'=0$  que es impar. Por tanto,

$$P = Z \oplus P' = ((A \oplus B) \oplus (C \oplus D)) \oplus P' \quad [5.17]$$

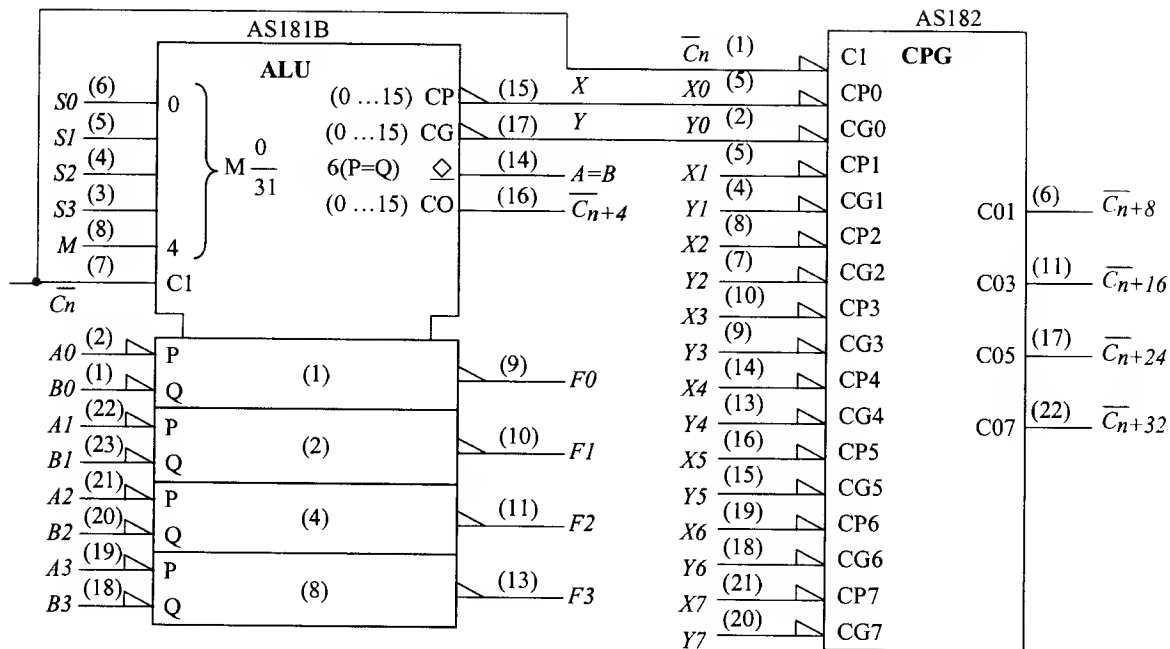
Es decir,  $P=1$  (*par*) si  $P'=0$  (*impar*) y  $Z=1$  (*impar*) y si  $P'=1$  (*par*) y  $Z=0$  (*par*). En cambio,  $P=0$  (*impar*) para  $P'=0$  y  $Z=0$ , y para  $P'=1$  y  $Z=1$ .



**Figura 5.21.** Síntesis de un detector de paridad para 4 bits.

## 5.5. UNIDADES ARITMÉTICO-LÓGICAS (ALU'S)

El circuito MSI más completo dentro del grupo de operadores aritmético-lógicos es la ALU (SN74181) (*figura 5.22*). Existen soluciones integradas en tecnología TTL, ECL y CMOS y contiene la síntesis de las funciones lógicas y aritméticas más usadas junto con 4 bits para la selección de función. La tabla de verdad de la *figura 5.23* muestra, para lógica positiva, las posibles funciones del circuito 181 y el valor de las variables de control que las seleccionan.



**Figura 5.22.** ALU, SN74AS181A. Símbolo lógico y conexión con el circuito AS882 (generador de acarreo adelantado) para aplicaciones de más bits que necesiten varias ALU's en cascada.

Esta unidad aritmético-lógica acepta como operandos dos palabras de 4 bits y un acarreo,  $C_n$ , para facilitar la conexión en cascada, en el caso de que necesitemos operar sobre palabras de más de 4 bits.

La primera selección de función se realiza a través de la variable de control  $M$  que permite elegir entre dos modos de funcionamiento:

$M=1 \Rightarrow$  Funciones lógicas.

$M=0 \Rightarrow$  Operaciones aritméticas.

Tras la selección del modo lógico opera la segunda columna de la tabla de la *figura 5.22*. Es decir, estando  $M$  en alta los cuatro bits de selección de función van a activar una de las 16 funciones lógicas de la tabla en cuestión.

Cuando  $M$  está en baja la ALU se configura como un procesador aritmético en el que a su

vez podemos decidir si se toma en cuenta o no el acarreo. Cuando no queremos considerar el arrastre ( $\overline{C_n}$  en alta), opera la tercera columna de la tabla. Finalmente, con arrastre opera la última columna.

SELECCIÓN					DATO ACTIVO EN ALTA		
S <sub>3</sub> S <sub>2</sub> S <sub>1</sub> S <sub>0</sub>	M=H	M=L Operaciones Aritméticas					
	Funciones Lógicas	$\overline{C_n}=H$ (sin acarreo)	$\overline{C_n}=L$ (con acarreo)				
L L L L	$F = \overline{A}$	$F = A$	$F = A \text{ PLUS } 1$				
L L L H	$F = \overline{A+B}$	$F = A + B$	$F = (A + B) \text{ PLUS } 1$				
L L H L	$F = \overline{A} B$	$F = A + \overline{B}$	$F = (A + \overline{B}) \text{ PLUS } 1$				
L L H H	$F = 0$	$F = \text{MINUS } 1 \text{ (comp. a 2)}$	$F = \text{Cero}$				
L H L L	$F = \overline{A} \overline{B}$	$F = A \text{ PLUS } \overline{A} \overline{B}$	$F = A \text{ PLUS } \overline{A} \overline{B} \text{ PLUS } 1$				
L H L H	$F = \overline{B}$	$F = (A + B) \text{ PLUS } \overline{A} \overline{B}$	$F = (A + B) \text{ PLUS } \overline{A} \overline{B} \text{ PLUS } 1$				
L H H L	$F = A \oplus B$	$F = A \text{ MINUS } B \text{ MINUS } 1$	$F = A \text{ MINUS } B$				
L H H H	$F = A \overline{B}$	$F = A \overline{B} \text{ MINUS } 1$	$F = A \overline{B}$				
H L L L	$F = \overline{A} + B$	$F = A \text{ PLUS } AB$	$F = A \text{ PLUS } AB \text{ PLUS } 1$				
H L L H	$F = \overline{A \oplus B}$	$F = A \text{ PLUS } B$	$F = A \text{ PLUS } B \text{ PLUS } 1$				
H L H L	$F = B$	$F = (A + \overline{B}) \text{ PLUS } AB$	$F = (A + \overline{B}) \text{ PLUS } AB \text{ PLUS } 1$				
H L H H	$F = A B$	$F = A B \text{ MINUS } 1$	$F = A B$				
H H L L	$F = 1$	$F = A \text{ PLUS } A$	$F = A \text{ PLUS } A \text{ PLUS } 1$				
H H L H	$F = A + \overline{B}$	$F = (A + B) \text{ PLUS } A$	$F = (A + B) \text{ PLUS } A \text{ PLUS } 1$				
H H H L	$F = A + B$	$F = (A + \overline{B}) \text{ PLUS } A$	$F = (A + \overline{B}) \text{ PLUS } A \text{ PLUS } 1$				
H H H H	$F = A$	$F = A \text{ MINUS } 1$	$F = A$				

**Figura 5.23.** Tabla de verdad para lógica positiva de la ALU 181.

Sea cual fuere el modo seleccionado y el valor de la variable de arrastre, las salidas están disponibles en  $F = F_3F_2F_1F_0$  y en el acarreo  $C_{n+4}$ . Así por ejemplo, si  $M=0$  y  $S=1001 \Rightarrow F=A+B$ , teniendo en cuenta el arrastre de etapas previas,  $C_n$ . En la tabla de verdad se utilizan las

palabras *plus* y *minus* para adición y resta aritmética, + para OR y  $\oplus$  para OR exclusivo.

Cuando la ALU se usa en adición y la velocidad es importante podemos desear un puente para el arrastre (circuito tipo 10179). Pensando en esto la ALU tiene dos terminales *P* y *G*. Una entrada de arrastre que vaya a propagarse a través de la ALU aparecerá en *P*. En cambio, un arrastre de salida generado dentro de la ALU aparecerá en *G*.

Este tipo de circuito puede considerarse como un "precursor" de la lógica programable en el sentido que la estudiaremos en el capítulo 7 ya que incluye las puertas lógicas necesarias para realizar varias funciones aritméticas y lógicas, junto con los circuitos de control necesarios para seleccionar las conexiones específicas para cada una de ellas. Las funciones que pueden realizar son del siguiente tipo:

*Aritméticas:* Suma, resta, comparación, desplazamiento de una posición del operando, etc...

*Lógicas:* OR exclusivo, comparación, AND, NAND, OR, NOR y 10 funciones más.

Cuando se realizan funciones aritméticas, deben facilitarse los arrastres internos, aplicando un "0" a la entrada de control de modo (*M*). Los terminales 15 y 17 se usan para generación simultánea de arrastre para aumentar la rapidez de operación. Entonces, la ALU debe usarse junto con el circuito SN74182, (generador de arrastre hacia adelante), del que hemos hablado anteriormente. Si no es importante la alta velocidad pueden usarse las entradas ( $C_n$ ) y salidas ( $C_{n+4}$ ) de arrastre. El circuito 74181 puede actuar en lógica positiva (activa en alta) o en lógica negativa (activa en baja) de acuerdo con la forma de usar sus terminales. La tabla que especifica la función realizada por la ALU y mostrada en la *figura 5.21* corresponde al criterio de lógica positiva (activa en alta).

La sustracción se realiza por suma en complemento a 1, generándose internamente el complemento a 1 del sustraendo. La salida resultante es:  $A-B-1$ , la cual requiere un "acarreo forzado" para proporcionar  $A-B$ .

La ALU también puede actuar como comparador. La salida  $A=B$  es decodificada internamente a partir de las funciones de salida  $F_0, F_1, F_2, F_3$ , de forma que cuando se aplican dos palabras de igual magnitud a las palabras *A* y *B* la comparación pasa a alta para indicar que son iguales.

La ALU también podría estar en modo de sustracción con  $C_n=1$  cuando se está realizando la comparación descrita anteriormente. La salida  $A=B$  es colector abierto de forma que admite la función AND cableada para ampliar la comparación a palabras de más de 4 bits. La salida de arrastre ( $C_{n+4}$ ) puede usarse también para obtener información sobre magnitudes relativas. De nuevo, la ALU podría situarse en el modo de sustracción cambiando los bits que controlan la selección de la función a la configuración 0110.

Estos circuitos están diseñados para incorporar además 16 funciones lógicas de dos variables sin el uso de ningún circuito externo adicional. En este caso se pone  $M=1$ , lo que inhibe el arrastre interno y se usan las entradas de selección,  $S_0S_1S_2S_3$ , para fijar la función lógica.

**Ejercicio:** *Al estudiar las distintas funciones aritméticas y lógicas que realiza la ALU, algunos alumnos se preguntan la diferencia que existen entre ciertas funciones como por ejemplo, entre:*

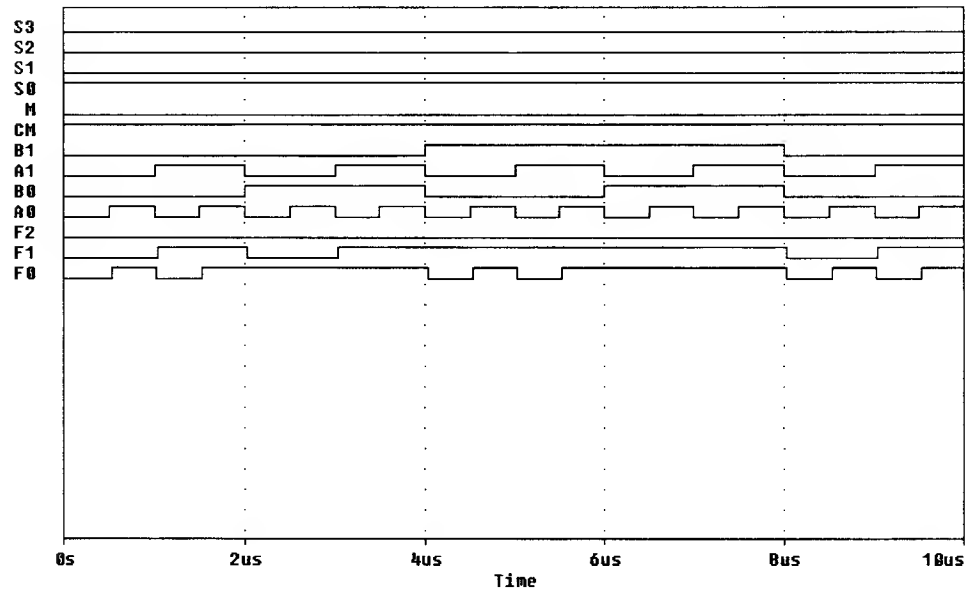
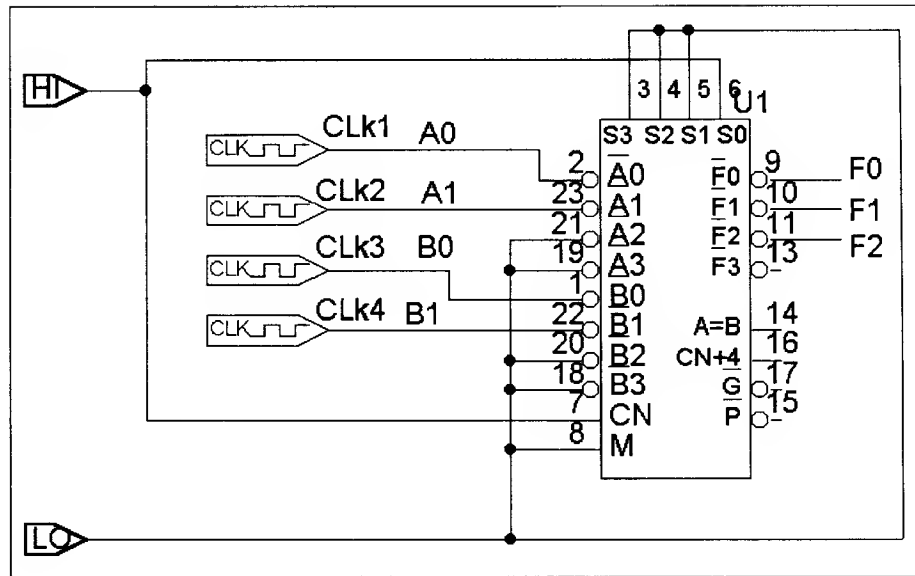
- (1)  $F=A+B$  considerada como operación aritmética
- (2)  $F=A$  plus  $B$ , aritmética
- (3)  $F=A + B$  , como operación lógica

*Pues bien, el signo +, tanto en la ALU seleccionada para realizar operación lógica como para realizar operación aritmética, es decir en los casos (1) y (3) significa la operación OR entre la palabra A y la B, mientras que la función plus es la suma aritmética.*

### ***Solución***

La figura 5.24 muestra el esquema y los resultados de la simulación. Veamos el primer caso. Es decir,  $F=A+B$  aritmético. En este caso, deberemos poner las señales de control en los siguientes valores:

S3	S2	S1	S0	M	$c_n$
L	L	L	H	L	H



**Figura 5.24.** Esquema de conexión en el simulador de la ALU SN74181 y resultado correspondiente en forma de cronograma que muestra la evolución temporal de las entradas y salidas de la ALU manteniendo constantes los valores de las señales de control que seleccionan la operación.

Para simplificar y facilitar el estudio hemos simplificado la ALU dejándola operativa para palabras de 2 bits en vez de usar los 4 bits que posee. Para ello los bits más significativos los hemos de poner a “0”. Además, para simular de forma dinámica las entradas y poder obtener la tabla de verdad a partir del diagrama de tiempos, hemos usado como entradas cuatro relojes que nos generan pulsos uno el doble del anterior. Es decir, para la entrada  $A0$  usamos un reloj digital de onda cuadrada y lo definimos para que la señal esté  $0.5\mu s$  en alta y  $0.5\mu s$  en baja. Para  $A1$  el periodo del reloj debe ser el doble, es decir lo definimos para que esté  $1\mu s$  en alta y  $1\mu s$  en baja. Para  $B0$  será el doble de  $A1$  y para  $B1$  el doble de  $B0$ .

A partir de este diagrama de tiempos podemos construir la tabla del funcionamiento de la ALU para esta configuración, tal como se muestra en la figura 5.25 y a continuación calculamos las expresiones de  $F2$ ,  $F1$  y  $F0$ . Como podemos observar,  $F2 = 0$ ,  $F1 = A1 + B1$  y  $F0 = A0 + B0$  luego la función resultante es la función OR entre las dos palabras  $A$  y  $B$ . Los apartados b y c de este ejercicio se resuelven de forma análoga.

$B1$	$A1$	$B0$	$A0$	$F2$	$F1$	$F0$
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	1	0	0	0	1	0
0	1	0	1	0	1	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	1	0	0	1	1
0	1	1	1	0	1	1
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	1	0	0	0	1	0
1	1	0	1	0	1	1
1	0	1	0	0	1	1
1	0	1	1	0	1	1
1	1	1	0	0	1	1
1	1	1	1	0	1	1

$B1 A1$				
$B0 A0$	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$\overline{F0} = \overline{A0} \overline{B0}$   $F0 = A0 + B0$

$B1 A1$				
$B0 A0$	00	01	11	10
00	0	1	1	1
01	0	1	1	1
11	0	1	1	1
10	0	1	1	1

$F1 = A1 + B1$

**Figura 5.25.** Obtención de las expresiones de  $F0$ ,  $F1$  y  $F2$ . Obsérvese que efectivamente, la función resultante es  $A$  OR  $B$ .



La auténtica potencia de este tipo de circuitos aritmético-lógicos se pone de manifiesto cuando se enlaza con otros circuitos de decodificación de instrucciones y de control de su ejecución de forma que es posible realizar cualquier secuencia de estas operaciones aritmético-lógicas, usando además los resultados de una etapa del proceso para decidir la operación a realizar en la etapa siguiente. Si añadimos la posibilidad de usar memorias RAM, tenemos la estructura básica de un sistema de cálculo digital, cuyo núcleo es la ALU. A ella se llevan desde la entrada o desde memoria los *operandos*. De ella se devuelven, a la salida o a otras posiciones de memoria, los *resultados* de la operación aritmética o lógica realizada. Finalmente, el "programa" es la secuencia de palabras de control que selecciona en cada momento la operación de la ALU, junto con el resto de los procesos de transferencia de datos, lectura o escritura en memoria y conexión con el exterior.

Junto a las funciones aritmético-lógicas consideradas hasta este momento (sumadores, restadores, comparadores, generadores-detectores de paridad y ALU's) existen otras funciones de procesado de información más específicas tales como los acumuladores (LS681), los procesadores "bit-slice" (AS888) y los llamados de aritmética de computación avanzada, relacionados con los procesos de multiplicación y división binarias (MC 10183, LS261). Es obvio que estos procesos pueden realizarse de forma algorítmica, vía programación, sin embargo, existen aplicaciones tales como la comunicación digital que exigen tan alta velocidad que sólo es posible ofrecerla mediante multiplicadores paralelo (SN74285, 74S558) sintetizados a nivel de circuito lógico de propósito especial. Su estudio, sin embargo, queda fuera del alcance de este curso.

También quedan fuera del propósito de este curso los microprocesadores y todos los circuitos cuyo nivel de complejidad los hace merecedores de asignaturas específicas o de capítulos especiales en cursos sobre arquitectura de ordenadores ó procesadores digitales de señal, por ejemplo.

En este tema de introducción al análisis y síntesis de funciones en lógica combinacional para operaciones aritmético-lógicas, al igual que en el resto de los temas dedicados al estudio de funciones combinacionales o secuenciales adecuadas al nivel de integración en media escala (MSI), el propósito pedagógico básico es ilustrar el tipo de función y el uso que se hace de los operadores del nivel inferior (puertas lógicas AND, OR, NAND ó NOR) para obtener funciones

específicas. No es sin embargo especialmente importante el ser exhaustivo describiendo todas y cada una de las funciones del nivel superior. Si el alumno ha adquirido las habilidades para el análisis y síntesis de sumadores con arrastre adelantado, por ejemplo, no tendrá dificultad en abordar el análisis o síntesis de otras funciones a partir de su circuito equivalente o de la descripción clara, completa, precisa e inequívoca de la función que se quiere sintetizar.

## 5.6. PROBLEMAS

**E.5.1.** *Las figuras 5.5 y 5.6 muestran los circuitos semisumador y sumador completo realizados con puertas AND y OR. Obtener las versiones equivalentes usando sólo puertas NAND.*

**E.5.2<sup>(\*)</sup>.** *El mismo desarrollo propuesto para explicar los circuitos sumadores completos a partir de semisumadores puede realizarse para el estudio de los circuitos restadores. Vamos a comprobarlo en este ejercicio. Las reglas para restar dos bits ( $A-B$ ) produciendo un bit de diferencia ( $D$ ) y un acarreo ( $C$ ) son las de la tabla adjunta.*

- ¿Cuales son las funciones lógicas del semirestador que genera  $C$  y  $D$  a partir de  $A$  y  $B$ ?*
- ¿Cómo podemos obtener un restador completo, (FS) a partir de dos semirestadores?*
- ¿Cómo deben de conectarse distintos FS para obtener un restador paralelo de cuatro bits?*

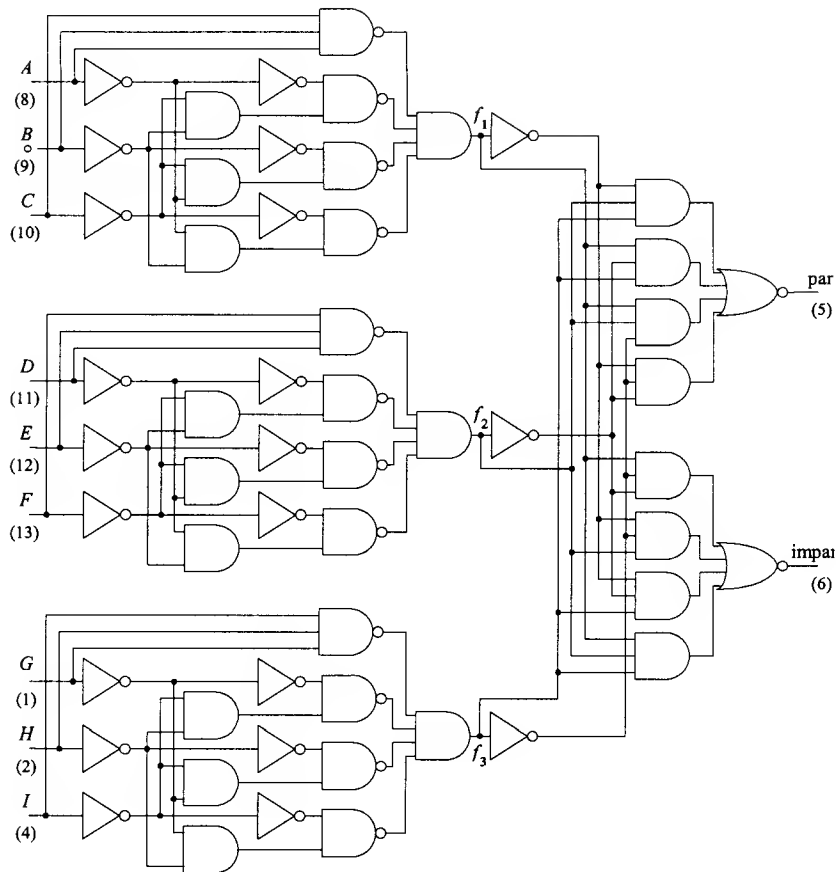
*Recuérdese que cuando se transfiere el acarreo de una etapa a la siguiente, se resta el sustraendo del minuendo, disminuyendo a continuación el resultado en 1.*

A	B	C	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

---

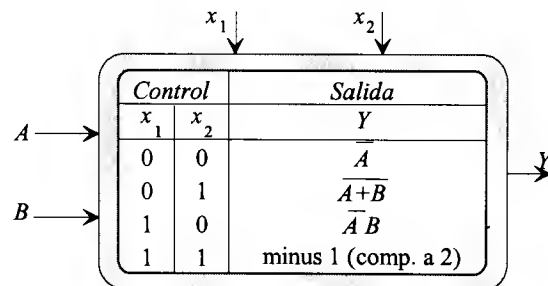
<sup>(\*)</sup> Este problema ha sido incluido en el material teórico del tema en esta segunda edición. Dejamos aquí también el enunciado porque hay un texto de problemas asociado a los enunciados de la primera edición.

- E.5.3.** *¿Cómo podrías realizar un circuito que fuera a la vez sumador y restador, tipo MC10180?. Recordemos que una resta es una suma en la que se ha cambiado el signo del sustraendo (complemento a 2). Recordemos que en la representación por complemento a 2 los números positivos se representan colocando un cero a la izquierda de los bits que representan la magnitud y los números negativos se obtiene a partir de los positivos negando todos los bits y sumando 1 al resultado. Este proceso de negar todos los bits y sumar un 1 al resultado se llama obtención del complemento a 2 del número. Esta representación de los números negativos es ventajosa a la hora de realizar operaciones aritméticas. La suma de números en complemento a 2 se obtiene a partir de una misma operación de suma sobre todos los bits de los operandos. Sólo puede haber rebose ("overflow"), cuando los números son ambos positivos o ambos negativos y esta situación se detecta comprobando el signo del resultado. Si es correcto, no hay rebose.*
- E.5.4.** *En las figuras 5.6 y 5.17 hemos visto la comparación de dos palabras de un bit y la extensión a cuatro bits. Supongamos ahora que disponemos de cinco comparadores de cuatro bits ( $A_0...A_3$ ), ( $B_0...B_3$ ). ¿Cómo deberíamos conectarlos para obtener un comparador para palabras de 16 bits?.*
- E.5.5.** *El circuito de la figura (SNALS280) es un generador/detector de paridad para nueve líneas de datos. Analizar las funciones lógicas para las tres vías ( $A, B, C$ ), ( $D, E, F$ ) y ( $G, H, I$ ) y obtener la ecuación que define la paridad (terminal 5 en alta y terminal 6 en baja. Esta respuesta debe darse para 0, 2, 4, 6 ó 8 líneas en alta. Probar por ejemplo para dos líneas. Inversamente, para que el terminal 6 esté en alta, el número de líneas de datos en alta deben ser, 1, 3, 5, 7 ó 9. Probar, por ejemplo, para una línea en alta.*

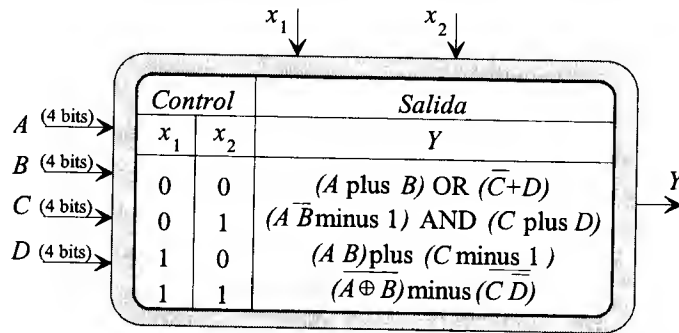


**E.5.6.** En la figura 5.23 se muestra la tabla de verdad de la ALU (AS181) junto con el valor de las variables de control ( $S_3$ ,  $S_2$ ,  $S_1$  y  $S_0$ ) y la selección de modo. Se dispone de tres ALU's y de cuantas puertas NOR se consideren necesarias. Diseñar a partir de estas componentes los siguientes circuitos:

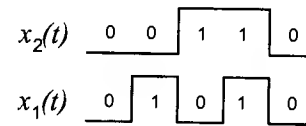
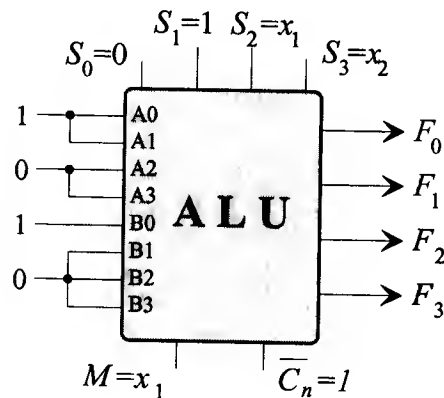
a)



- b) Considerando ahora cuatro palabras de cuatro bits ( $A0...A3$ ;  $B0...B3$ ;  $C0...C3$ ;  $D0...D3$ ) como datos de entrada y las mismas dos variables de control, ( $X1$ ,  $X2$ ), diseñar un circuito que realice la siguiente función:



- c) Supongamos que podemos disponer de una secuencia de control en la que las dos variables, ( $X1$  y  $X2$ ) recorren de forma cíclica sus cuatro valores posibles (00, 01, 10, 11) (veremos más adelante que esta secuencia se consigue de forma muy sencilla mediante un contador binario), ¿qué forma de onda se obtiene a la salida ( $F0$ ,  $F1$ ,  $F2$ ,  $F3$ ) del circuito de la figura?



---

# ***Preparación de la Evaluación***

---

*Recordemos que hemos descompuesto el contenido de este tema en 4 objetivos. Por consiguiente, la ayuda a la evaluación también la vamos a descomponer de acuerdo con esos objetivos.*

## **Objetivos**

- 1:** *Aritmética Binaria*
- 2:** *Circuitos Sumadores y Restadores*
- 3:** *Circuitos Comparadores*
- 4:** *ALU's*

## **1. Aritmética Binaria**

**1.1.** *Describa las ventajas e inconvenientes de los distintos tipos de representación de números negativos.*

**1.2.** *Usar la tabla en la que aparecen las distintas representaciones (binario puro, S-M, C-1 y C-2) para resolver las siguientes cuestiones:*

**a)** *Sumar en S-M los siguientes números:*

**a.1)**  $1010 + 0101$

**a.2)**  $1111 + 1101$

*¿Qué problema ha encontrado?. ¿Cuál es el mayor número decimal que puede representarse usando palabras de 4 bits?.*

**b)** *Realice las operaciones de suma y resta de dos palabras de 4 bits en las tres representaciones (S-M, C-1 y C-2). Comente los resultados.*

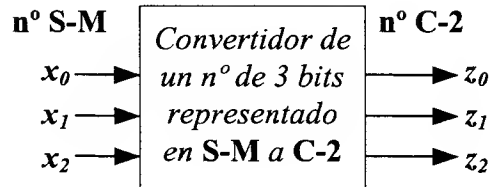
**b.1)**  $4 + 3$

**b.2)**  $-4 + 5$

b.3) -4 - 5

b.4) 4 - 5

1.3. Diseñar el circuito combinacional necesario para hacer automáticamente la conversión de S-M a C-2



1.4. Proponer un circuito combinacional que convierta representaciones de C-1 a C-2.

## 2. Circuitos Sumadores y Restadores

2.1. Recordando los posibles valores de la suma de dos dígitos ( $0+0=0$ ,  $0+1=1+0=1$ ,  $1+1=10$ ) y de tres dígitos binarios ( $0+0+0=0$ ,  $0+0+1=1$ , ...,  $1+1+1=11$ ). ¿Podría describir con palabras la función de acarreo (C) y la función de la suma (S) de un semisumador y de un sumador?

### Solución

De la simple observación de los valores anteriores es fácil obtener las siguientes descripciones

a) **Semisumador:** La función suma es cero ( $S=0$ ) cuando los dos bits son iguales ( $A=B=0$  ó  $A=B=1$ ) y es uno ( $S=1$ ) cuando los bits son diferentes ( $A \neq B$ ).

La función de acarreo es uno ( $C=1$ ) cuando ambos bits están en alta ( $A=B=1$ )

Por consiguiente, el bit de suma es el *OR exclusivo* y el de acarreo el *producto lógico*.

b) **Sumador:** Cuando se considera el acarreo de la suma de bits anteriores, la función suma (S) es una función de tres variables. Está en alta cuando el número de bits de los operandos y el acarreo que están en alta es impar (uno o tres). Por tanto, la suma es el *OR exclusivo* de las entradas y el acarreo de entrada.

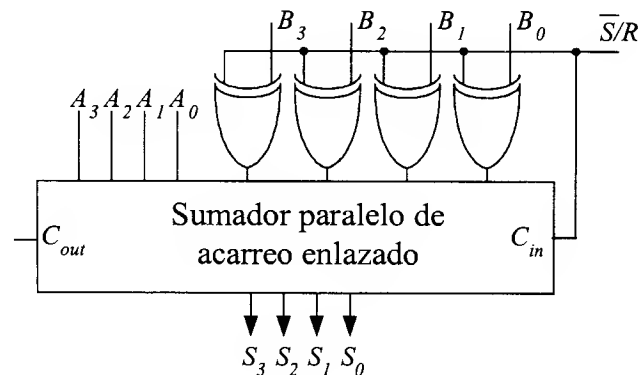
La función de acarreo ( $C$ ) está en alta cuando el número de bits en alta es par o bien los tres están en alta.

*Dejamos para el alumno las descripciones equivalentes en el caso del restador*

**2.2.** Supongamos que ya sabemos diseñar sumadores de un bit. Sin embargo, lo que necesitamos es saber sumar palabras de  $n$  bits a partir de  $n$  circuitos que sumen palabras de un bit. La forma más sencilla de conseguir esto es conectar los  $n$  sumadores de 1 bit enlazados por los acarrees. Al primer sumador entran los bits menos significativos de ambas palabras ( $A_0, B_0$ ) y al último los más significativos ( $A_3, B_3$ , para palabras de 4 bits). El primer acarreo,  $C_0$ , lo consideramos como el acarreo de entrada al circuito ( $C_{in}$ ). Los siguientes acarrees son los acarrees intermedios y el de los bits más significativos ( $C_{n+1}$ ) lo consideramos como acarreo de salida del circuito ( $C_{out}$ ).

- a) ¿Podría dibujar el circuito correspondiente a un sumador de este tipo para palabras de 4 bits?
- b) Supongamos ahora que este sumador de 4 bits lo convertimos en un módulo de diseño. ¿Cómo conectaría dos de estos módulos para obtener un sumador de 8 bits?
- c) ¿Encuentra alguna fórmula recursiva para pasar de sumadores paralelo con acarreo enlazado de  $n$  bits a sumadores de  $kn$  bits ( $K = 2, 3, 4, \dots$ )?. Dibújela.

**2.3.** Analizar el circuito de la figura. ¿Qué función realiza cuando la entrada  $\overline{S}/R$  está en alta?. ¿Y cuándo está en baja?





- 2.4. *¿Cómo podríamos conectar dos sumadores de acarreo adelantado de 4 bits para conseguir uno de 8 bits?.*
- 2.5. *Diseñar un sumador/restador paralelo de 2 palabras de 4 bits en C-1 con propagación del acarreo en serie que detecte el rebose.*
- 2.6. *Diseñar un sumador/restador paralelo de 2 palabras de 4 bits en C-2 con propagación del acarreo en serie que detecte el rebose.*
- 2.7. *Estime el retardo de un sumador de cuatro bits con acarreo adelantado. Recuerde que para evaluar el retardo en la parte del circuito que genera el acarreo adelantado se puede usar, por ejemplo, el circuito figura 5.12. Se recomienda para el cálculo utilizar la situación correspondiente al “peor de los casos”, (1111+0001) donde cada bit de suma genera un acarreo hacia el siguiente.*
- 2.8. *En la figura 5.15 del texto se ha presentado la solución a la suma por C-1 incluyendo el circuito detector de rebose. Repite ahora el procedimiento equivalente para aritmética en Complemento a 2. Es decir,*
- a) *describa en extenso (tablas de verdad) el procedimiento de sumar dos números positivos y negativos en C-2.*
  - b) *Diseñe el circuito sumador correspondiente distinguiendo las partes correspondientes a la suma binaria y a la detección de error por rebose.*

*Aconsejamos usar una señal de control para decidir si el circuito debe sumar o restar ( $C=1$  suma y  $C=0$  resta). Recuerde que para restar se complementa a 2 el sustraendo. Conviene recordar también las condiciones de rebose (si los dos operandos son positivos y la suma sale negativa y si los dos operandos son negativos y la suma es positiva)*

### **3. Circuitos Comparadores**

- 3.1. a) *Escriba las funciones de un comparador para palabras de 2 bits.*
- b) *¿Cómo podría ampliarlo a palabras de 3 bits?. ¿Qué esquema de conexión se le ocurre para pasar de  $n$  a  $(n+1)$  bits?. ¿Podría relacionar este ejercicio con el problema 5.4 del texto de problemas?.*

- 3.2. *Hasta ahora los comparadores sólo hacían referencia a las magnitudes. ¿Podría diseñar dos comparadores para palabras de 2 bits en cada una de las representaciones S-M y C-2?*
- 3.3. *Diseñar un comparador de 2 palabras de 3 bits con el circuito de control correspondiente para que a la salida muestre la mayor de las dos palabras que compara.*
- 3.4. *Diseñar un comparador para dos palabras de 3 bits ( $A_2, A_1, A_0$ ;  $B_2, B_1, B_0$ ) en representación Magnitud-Signo que proporcione tres salidas ( $A > B$ ,  $A = B$ ,  $A < B$ ).*

#### 4. ALUs

- 4.1. *En la figura 5.23 se muestra la tabla de operación de la ALU SN74181 para lógica positiva. Especificar las funciones que realiza este circuito si:*

a)  $S = L H H H, M = H \text{ y } \overline{c_n} = H$

b)  $S = L L L H, M = L \text{ y } \overline{c_n} = L$

- 4.2. *Manejar la tabla completa de la ALU es tedioso por lo que es aconsejable hacer ejercicios sobre “mini-ALUs”. Es decir, sobre versiones reducidas que sólo usan dos variables de control (por ejemplo) y, por consiguiente, sólo nos permiten seleccionar entre cuatro operaciones. Revisar el problema E.5.6 del texto de problemas y, usándolo como inspiración:*

- a) *Diseñar un circuito de control que permita seleccionar entre las funciones Lógicas:*

$$F = A$$

$$F = \overline{A + B}$$

$$F = \overline{A} \cdot B$$

$$F = 0$$

- b) *Diseñar otro circuito de control que permita seleccionar entre las funciones Lógicas:*

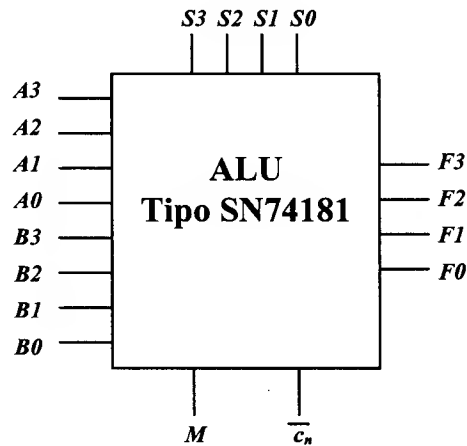
$$F = \overline{A \cdot B}$$

$$F = \overline{B}$$

$$F = A \oplus B$$

$$F = A \cdot \overline{B}$$

**4.3.** Las ALU's son circuitos que realizan funciones múltiples controlables en las que, de acuerdo con el valor de ciertas variables de control ( $S_0$ ,  $S_1$ ,  $S_2$ , ...,  $M$  y  $c_n$ ) se realizan distintas operaciones aritméticas o lógicas sobre las dos palabras de entrada ( $A_0 A_1 A_2 \dots A_n$ ) y ( $B_0 B_1 B_2 \dots B_n$ )

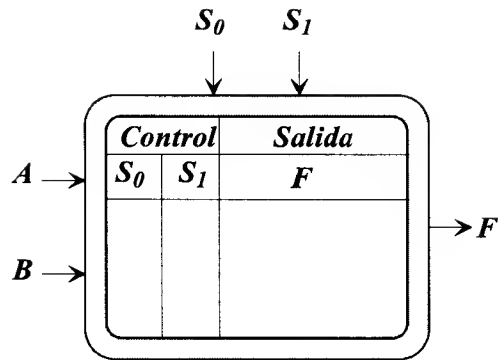


En la figura 5.19 se muestra la tabla de operación de la ALU SN74181 para lógica positiva. Especificar las funciones que realiza este circuito si:

- a)  $S = H L L L$ ,  $M = L$  y  $\overline{c_n} = L$
- b)  $S = H H H L$ ,  $M = H$  y  $\overline{c_n} = H$
- c)  $S = H L L H$ ,  $M = H$  y  $\overline{c_n} = L$

Obsérvese que cuando la operación es lógica los acarreos no intervienen, por lo que su valor es irrelevante.

**4.4.** El esquema de la figura adjunta representa una “mini ALU” vacía. Seleccione cuatro funciones cualesquiera (dos aritméticas y otras dos lógicas) y diseñe el circuito de control correspondiente.



## REFERENCIAS BIBLIOGRÁFICAS

1. Dormido, S., Canto, M.A., Mira, J., Delgado, A.E.: “*Estructura y Tecnología de Computadores II*”. Sanz y Torres, 1994, 1999, 2000.
2. Gajski, .D.D.: “*Principios de Diseño Digital*”. Prentice Hall, 1998.Mira,
3. Katz, R.H.: “*Contemporary Logic Design*”. The Benjamin/Cummings Pub. Com., 1994.
4. Sandige, R.S.: “*Modern Digital Design*”. McGraw-Hill. 1990.
5. Texas Instruments: “*The TTL Data Book*”. 1987.



TEMA

6

---

## ***Lógica Combinacional (II): Ruta de datos***

---

### **+ CONTEXTO**

*Recordemos que hay esencialmente tres tipos de funciones en lógica combinacional:*

- a)** *Funciones de ruta de datos*
- b)** *Funciones aritmético-lógicas (ALU)*
- c)** *Circuitos cambiadores de código*

*Las primeras son las encargadas de guiar, de forma controlada, el tráfico de señales (datos e instrucciones) entre los distintos puntos de un sistema de cálculo.*

*Las segundas son las encargadas de realizar operaciones aritméticas o lógicas sobre los datos, transformándolos y produciendo un resultado local que, de nuevo, entra al tráfico controlado.*

*Ya hemos visto las operaciones aritmético-lógicas, los comparadores y la estructura interna de una ALU. En este tema estudiaremos las funciones de ruta de datos en lógica no programable.*

*Esencialmente los multiplexos y demultiplexos que están basados en la apertura y cierre de puertas bajo el control de los términos mínimos de un conjunto de variables de control.*

## **++ CONOCIMIENTO PREVIO NECESARIO**

*Todos los conocimientos previos necesarios sobre la representación, análisis, minimización y síntesis de funciones lógicas se encuentran en el tema 1*

## **+++ OBJETIVOS DEL TEMA**

**Objetivo 1:** *Comprender la estructura interna y la función de los multiplexos en su doble uso como:*

- a)** *Selector de canales de entrada (1 de n, paso de información de paralelo a serie)*
- b)** *Módulo de diseño de otros circuitos en lógica combinacional.*

**Objetivo 2:** *Comprender los demultiplexos y su uso como:*

- a)** *Selector de canales de salida (1 de n, paso de información de serie a paralelo)*
- b)** *Módulo de diseño (junto con los Multiplexos) de circuitos convertidores de código.*

**Objetivo 3:** *Conocer los circuitos encargados de adaptar las señales en su interacción con un “bus”: amplificadores (buffers-drivers) y transmisores-receptores de bus (bus-transceivers).*

## **++++ GUÍA DE ESTUDIO**

*La estrategia seguida para la presentación del material es estudiar primero las funciones para un número pequeño de bits y conectar después con los circuitos comerciales y con el uso de versiones más complejas ya integradas en arquitecturas de bus.*



# Contenido del Tema

## 6.1. CIRCUITOS SELECTORES DE DATOS (MULTIPLEXOS)

Empezaremos estudiando los multiplexos. Un multiplexo de  $N$  a 1 es un circuito combinacional que permite situar en un canal de salida la señal procedente de cualquiera de los  $N$  canales de entrada. Para la selección de canal hacen falta  $n$  variables binarias tales que  $2^n = N$ . Es decir, cada configuración lógica mutuamente exclusiva de esas variables de control abre un canal asegurando que durante el intervalo de tiempo en el que ese canal permanece abierto, los otros están cerrados.

Tal como se muestra en la figura 6.1, la forma natural de seleccionar cuál de dos señales de entrada ( $D_0$ ,  $D_1$ ) pasa a un canal de salida  $Y$  es llevarlas a dos puertas AND que se unen finalmente en una puerta OR. Después hay que añadir la señal de control ( $A$ ), de forma que de acuerdo con el valor de esa señal pase uno u otro canal de entrada. Por ejemplo, para  $A=0$ , pasa  $D_0$ , ( $Y= D_0$ ), y para  $A=1$  pasa  $D_1$ , ( $Y= D_1$ ). Tenemos así un circuito multiplexor de 2 a 1.

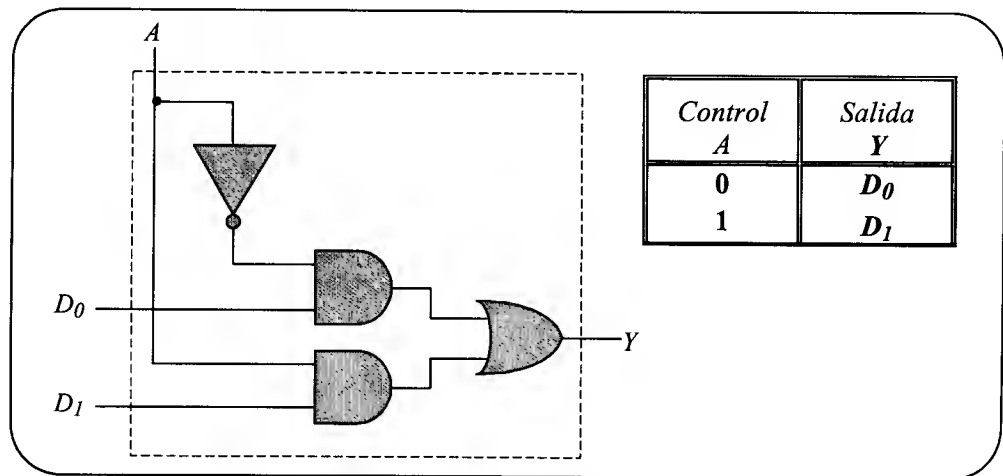
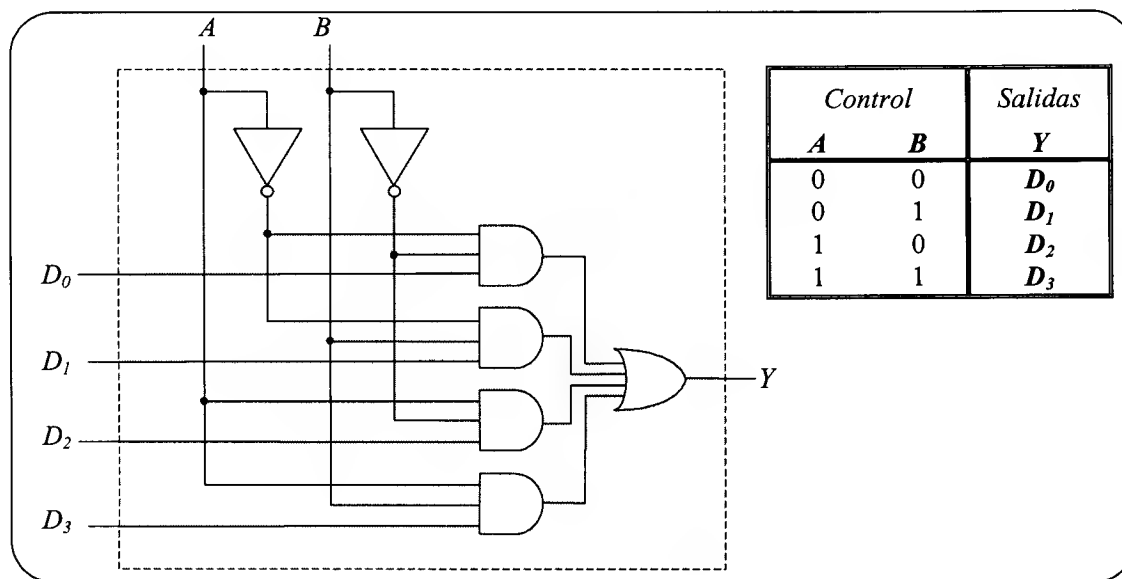


Figura 6.1. Circuito multiplexor de 2 a 1.

De forma análoga, para cuatro canales de entrada ( $D_0$ ,  $D_1$ ,  $D_2$ ,  $D_3$ ) necesitamos cuatro puertas

AND y cuatro señales de control que se generan con dos variables,  $A$  y  $B$ . Finalmente, volvemos a poner una puerta OR y tenemos un circuito multiplexor de 4 a 1, tal como se muestra en la *figura 6. 2*.

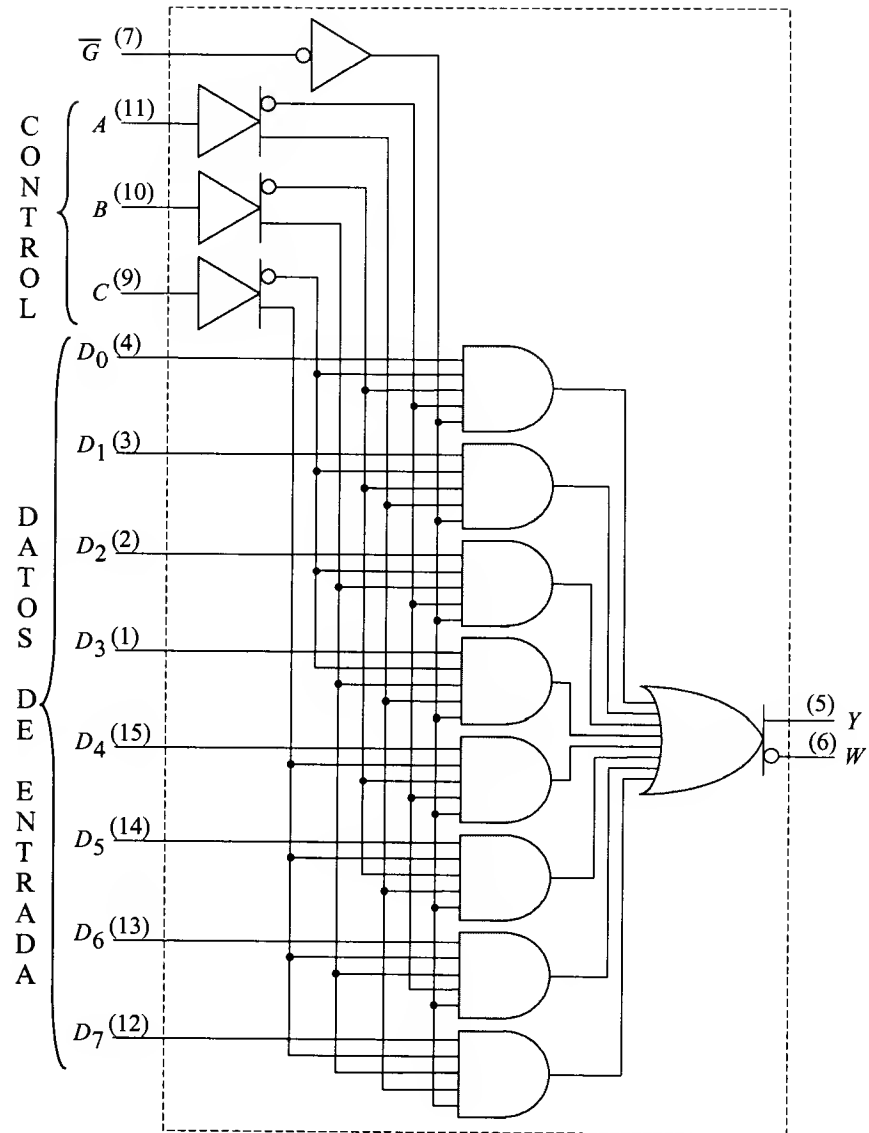


**Figura 6.2.** Circuitos multiplexores de 4 a 1.

Es fácil ahora generalizar a  $N$  canales de entrada. Basta repetir la estructura AND/OR y aumentar a  $n$  el número de líneas de control, siendo  $2^n=N$ . La *figura 6.3* muestra el circuito interno correspondiente al multiplexo de ocho líneas a una, AS151, y la 6.4 la tabla de verdad y el símbolo lógico correspondientes. El circuito posee ocho canales de entrada,  $(D_0, D_1, \dots, D_7)$ . Para controlar su selección hacen falta tres variables,  $A$ ,  $B$  y  $C$  (ya que  $2^3 = 8$ ). A partir de estas variables (y de sus inversas) se producen los ocho términos mínimos en la columna de puertas AND. Cada una de estas puertas posee un término mínimo y un canal  $D_i$  de forma que sólo pasa a la salida, a través de la puerta OR, el canal  $D_i$  tal que su subíndice en base dos coincide con la configuración de control. Así, para  $A = B = C = 0$ , pasa  $D_0$ ; para  $A = B = 0$  y  $C = 1$ , pasará  $D_4$  y así sucesivamente.

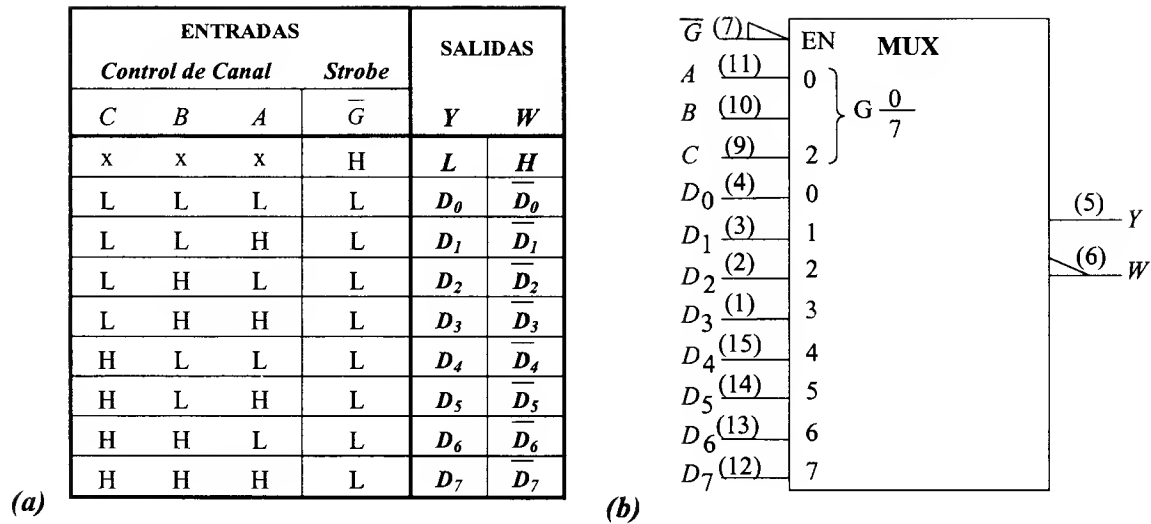
En general estos circuitos poseen una señal de inhibición/facilitación ("strobe") que controla el instante en el que se realiza la transferencia del dato. En el caso del circuito de la figura la señal de inhibición ( $\bar{G}$ ) debe estar en baja para facilitar el paso a  $Y$  de la señal  $D_i$  seleccionada. Cuando  $\bar{G}$  está en alta, la salida  $W$  está en alta y la  $Y$  en baja, independientemente del valor de las entradas.





**Figura 6.3.** Circuito interno del Multiplexo AS151.

Este circuito puede usarse no sólo para su función de selector de canal sino también para conversión paralelo a serie (con la apertura sucesiva de los distintos canales) y para la síntesis de funciones lógicas arbitrarias. Veamos por su importancia esta segunda aplicación.



**Figura 6.4.** . (a) Tabla de verdad del Multiplexo AS151. (b) Símbolo lógico.

Cuando aparece la necesidad de sintetizar una función lógica puede ocurrir que ya exista integrada en cuyo caso se usa directamente. Si la función que necesitamos no existe hay que sintetizarla usando puertas lógicas (nivel SSI), circuitos del nivel MSI (multiplexos, por ejemplo) o lógica programable, de la que hablaremos en el tema siguiente. Si recordamos el tema primero sobre las distintas forma canónicas en las que se puede representar una función lógica es fácil reconocer la idoneidad de los multiplexos para el diseño lógico. De hecho un multiplexo es la implementación en dos niveles (AND, OR) de una función lógica universal escrita en forma normal disyuntiva, es decir, como suma de productos. La salida es la suma de los productos de los términos mínimos de las variables de control de canal ( $m_0 = \overline{A} \overline{B} \overline{C}$ ,  $m_1 = \overline{A} \overline{B} C$ , ...,  $m_7 = A B C$ ) por los datos de entrada ( $D_0$ ,  $D_1$ , ...,  $D_7$ ) que realizan la función de los coeficientes de esos términos mínimos. Cuando un dato,  $D_i$ , está en alta decimos que la función posee el término mínimo correspondiente,  $m_i$ , de forma que si en ese momento apareciera en las entradas de control, la salida del multiplexo ,  $Y$ , estaría en alta,

$$Y = \left[ \sum_{i=0}^{2^n-1} D_i \cdot m_i \right] \cdot \overline{G} \quad [6.1]$$

Con un multiplexo de 8 entradas podemos generar cualquier función de cuatro variables. Para

funciones de tres variables necesitaríamos un multiplexo de sólo cuatro entradas. Veamos primero un ejemplo sencillo de diseño para funciones de tres variables. Supongamos que queremos realizar la función,

$$f(X,Y,Z) = \overline{X} \overline{Y} + \overline{X} Y Z + X \overline{Y} \overline{Z} + X Y Z \quad [6.2]$$

Vamos a hacer corresponder a las dos primeras variables  $(X,Y)$  con las dos señales de control  $(A,B)$  de un multiplexo de cuatro entradas. Con ello tendríamos una parte de cada uno de los términos mínimos. Para completarlos usamos las líneas de datos  $(D_0, D_1, D_2, D_3)$  que deben corresponder a los valores que toma la variable  $Z$ . En el caso del ejemplo el término mínimo  $(m_0 = \overline{X} \overline{Y})$  no posee  $Z$ , viene multiplicado por "1", luego tengo que poner  $D_0 = 1$ . El segundo término  $(m_1 = \overline{X} Y)$ , viene multiplicado por  $Z$ , luego pondremos  $D_1 = Z$ . El tercer término mínimo  $(m_2 = X \overline{Y})$  aparece multiplicado por  $\overline{Z}$ , luego  $D_2 = \overline{Z}$ . Finalmente,  $m_3 = X Y$ , y  $D_3 = Z$ . Evidentemente, este ejemplo ha sido elegido con propósito pedagógico, pero la regla general es sencilla.

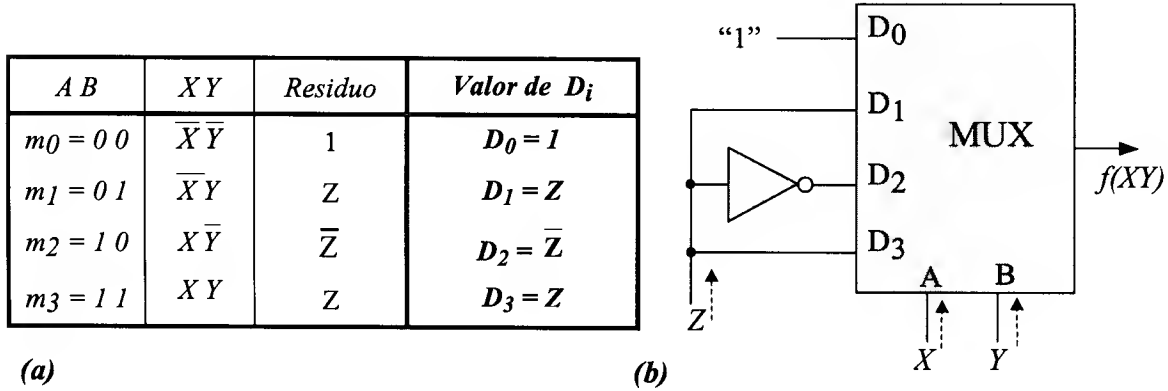
1. Las variables de control  $(A, B)$  representan a dos variables cualesquiera de la función a sintetizar.
2. Los canales de datos  $(D_0, D_1, D_2, D_3)$  se usan para la tercera variable  $(Z)$ . El rango de valores para los  $D_i$  es  $(0, 1, Z, \overline{Z})$ . Si un determinado producto  $X \cdot Y$  no existe, ponemos la  $D_i$  correspondiente a "0". Si existe y no tiene  $Z$ , ponemos su  $D_i$  a "1". Si tiene  $Z$ , ponemos el valor correspondiente  $(\overline{Z} \text{ ó } Z)$ .

Para sistematizar el proceso es conveniente comparar las tablas de verdad correspondientes, tal como se ilustra en la figura 6.5.

Cuando el número de variables es tal que un determinado tipo de multiplexo no es suficiente para su síntesis, se utiliza un diseño en árbol. Supongamos que queremos sintetizar una función de cinco variables,  $f(X, Y, Z, U, V)$ , para lo que necesitamos un multiplexo de 16 entradas y que sólo disponemos de multiplexores de 4 entradas. Veamos cómo podemos realizar la síntesis en dos niveles. Para ello realizamos una partición de las variables. Dejamos, por ejemplo  $(X,Y)$  para las entradas de

control del segundo nivel y sintetizamos el resto ( $Z$ ,  $U$  y  $V$ ) en el primer nivel.

Cuando algún término mínimo en  $(X,Y)$  no existe la  $D_i$  correspondiente se pone a "0". Cuando existe de forma aislada sin participación del resto de las variables,  $D_i$  se pone a "1". Finalmente, cuando hay participación de ( $Z$ ,  $U$ , ó  $V$ ), se saca factor común y se sintetizan en el primer nivel para actuar después como  $D_i$  del segundo. Este procedimiento de reparto de variables sobre distintos niveles es iterativo.



**Figura 6.5.** Diseño de funciones lógicas usando multiplexos. Ilustración del ejemplo usando un multiplexo de cuatro entradas ( $D_0$ ,  $D_1$ ,  $D_2$  y  $D_3$ ) y dos líneas de control ( $A$  y  $B$ ). **(a)** Tablas de la función a sintetizar y los valores de  $D_i$  para la variable residual. **(b)** Esquema de conexión del circuito.

Veamos el procedimiento de diseño con multiplexos descrito anteriormente de forma general sobre un ejemplo que incluya la mayoría de las circunstancias descritas anteriormente (ceros, unos y variables que pasan al nivel anterior). Supongamos que necesitamos sintetizar la función:

$$f(X,Y,Z,U,V) = \bar{X} \bar{Y} \bar{Z} \bar{U} V + \bar{X} \bar{Y} U \bar{V} + X \bar{Y} Z U \bar{V} + X \bar{Y} \bar{Z} U \bar{V} + X Y Z U V + X Y \bar{Z} \bar{U} \bar{V} \quad [6.3]$$

**Paso 1º:** Identificar los términos mínimos de  $(X,Y)$  y sacar factor común.

$$f(X,Y,Z,U,V) = \overbrace{\bar{X} \bar{Y} (\bar{Z} \bar{U} V + U \bar{V})}^{D0} + \overbrace{\bar{X} Y (0)}^{D1} + \overbrace{X \bar{Y} (Z U \bar{V} + \bar{Z} U \bar{V})}^{D2} + \overbrace{X Y (Z U V + \bar{Z} \bar{U} \bar{V})}^{D3} \quad [6.4]$$

Así, ya tenemos el *segundo nivel*.

**Paso 2º:** Sintetizar los paréntesis en el primer nivel. Síntesis de  $(\bar{Z}\bar{U}V + U\bar{V})$ . Asociamos  $U, V$  a  $A, B$  y dejamos  $Z$  para las  $D_i$ :

$$\begin{array}{ccccccc} & \xleftrightarrow{D_0} & & \xleftrightarrow{D_1} & & \xleftrightarrow{D_2} & & \xleftrightarrow{D_3} \\ (\bar{Z}\bar{U}V + U\bar{V}) = & \bar{U}\bar{V}(0) & + & \bar{U}V(\bar{Z}) & + & U\bar{V}(1) & + & UV(0) \end{array} \quad [6.5]$$

Análogamente,

$$\begin{array}{ccccccc} & \xleftrightarrow{D_0} & & \xleftrightarrow{D_1} & & \xleftrightarrow{D_2=1} & & \xleftrightarrow{D_3} \\ (ZU\bar{V} + \bar{Z}U\bar{V}) = & \bar{U}\bar{V}(0) & + & \bar{U}V(0) & + & U\bar{V}(Z + \bar{Z}) & + & UV(0) \end{array} \quad [6.6]$$

$$\begin{array}{ccccccc} & \xleftrightarrow{D_0} & & \xleftrightarrow{D_1} & & \xleftrightarrow{D_2} & & \xleftrightarrow{D_3} \\ (ZUV + \bar{Z}\bar{U}\bar{V}) = & \bar{U}\bar{V}(\bar{Z}) & + & \bar{U}V(0) & + & U\bar{V}(0) & + & UV(Z) \end{array} \quad [6.7]$$

Con lo que se obtiene el circuito que se muestra en la *figura 6.6*.

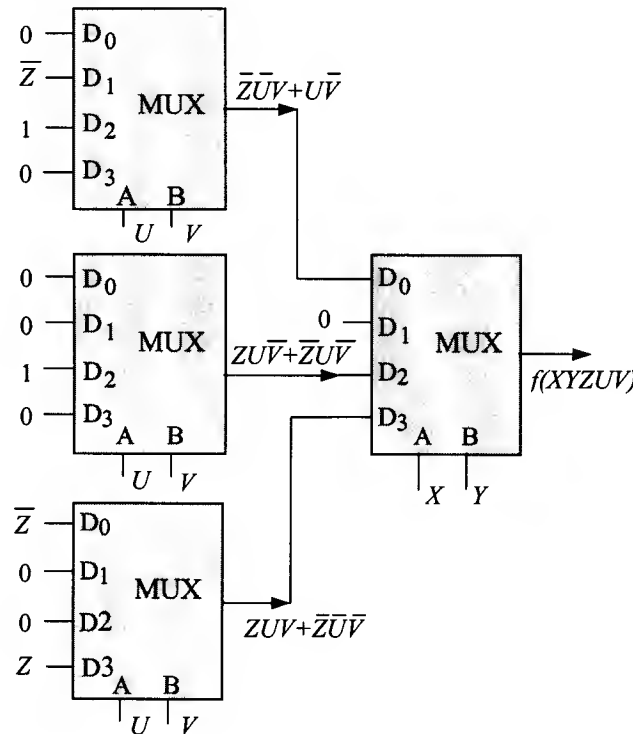
Hay otras muchas aplicaciones de los multiplexos asociadas a su función de selección de un camino entre  $N$  posibles. Por ejemplo, en todos los circuitos digitales en los que la comunicación se organiza usando canales de datos, a los que están conectados varios circuitos que pueden ser alternativamente *fuentes* o *destinos* de información. El facilitar la entrada y/o salida del canal de los datos procedentes de un circuito concreto siempre se puede realizar usando multiplexos y sus circuitos complementarios (demultiplexos) de los que hablaremos a continuación.

Para todas estas funciones específicas de multiplexado existen soluciones integradas en MSI. Por ejemplo, en TTL tenemos multiplexores de 16 a 1 con salidas de dos estados (SN74150) y de tres estados (SN74250) o de 8 a 1, también con salidas de dos (SN74151) o tres estados (SN74251) ó multiplexos universales tipo SN74857 en los que, además de seleccionar un canal de entrada, se ofrece a la salida alguna operación lógica entre las entradas.

En todos los casos, junto a las entradas de datos y los bits de control se incluye un terminal de facilitación del circuito (*enable*,  $\bar{G}$ ), activo en baja, que cuando está en alta pone la salida en estado de alta impedancia pudiendo así seguir conectada directamente al bus. También es usual que las salidas sean dobles, la correspondiente al canal de entrada seleccionado y su complementaria.

En los universales hay seis multiplexos de 2 a 1, con las funcionalidades usuales (tres estados,

salida real y complementada, etc.), pero con la facilidad adicional de admitir cuatro modos de operación: selecciona el bit  $A$  de cada par, selecciona el bit  $B$ , realiza la operación AND entre  $A$  y  $B$  o pasa todas las entradas a baja (*clear*).



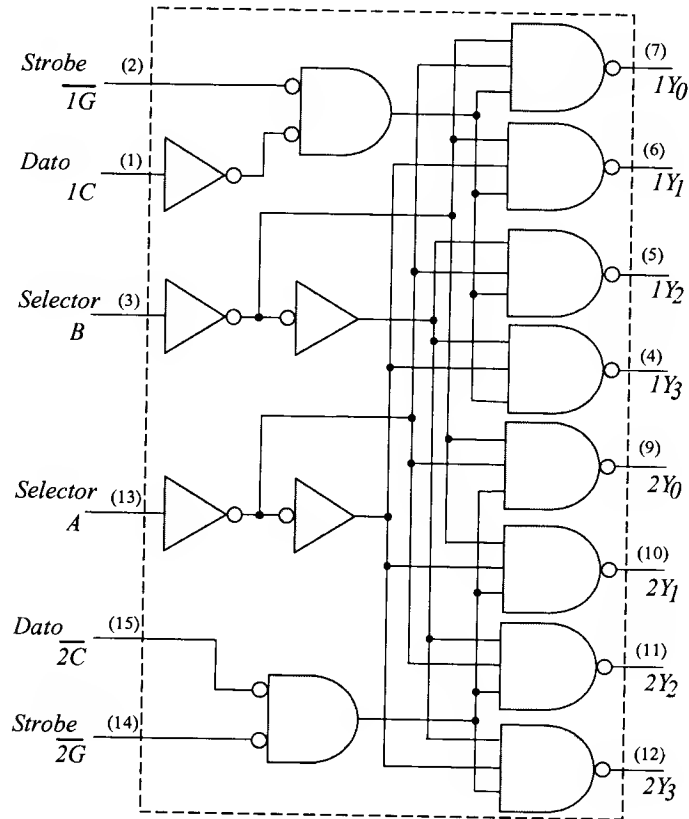
**Figura 6.6.** Diseño en dos niveles de una función de cinco variables usando MUX de cuatro entradas.

El diseño de funciones lógicas con multiplexos es un procedimiento sencillo y universal. Veremos su utilidad en los temas 7 y 13, al estudiar la lógica programable. Una parte importante del proceso de configuración de un circuito combinacional o secuencial se resuelve mediante el uso de multiplexos que, dependiendo del valor de sus líneas de control, configuran distintos esquemas de conectividad entre otros circuitos (puertas, contadores, registros , etc...).

## 6.2. DEMULTIPLEXOS.

Realizan la función inversa a la de los multiplexos. Un demultiplexo de 1 a  $2^n$  es un circuito con una única línea de entrada y  $2^n$  líneas de salida cuya puerta de acceso está controlada por  $n$

variables. Para cada configuración mutuamente exclusiva de estas líneas de control se abre un sólo canal de salida. La función se completa con la inclusión de una señal de facilitación o inhibición de la transmisión ("enable/strobe",  $E$ ). La *figura 6.7* muestra el circuito interno del SN74ALS156 que incluye dos demultiplexos de una línea de entrada a cuatro líneas de salida con líneas de inhibición individuales ( $\overline{1G}$  y  $\overline{2G}$ ), y selección de dirección de salida común a ambas partes del circuito (señales  $A$  y  $B$ ). Las entradas son  $1C$  y  $2C$ .



**Figura 6.7.** Circuito interno del SN74ALS156 que incluye dos demultiplexos 1 a 4 con direccionamiento común ( $A, B$ ) y entrada de facilitación/inhibición individual ( $\overline{1G}$  y  $\overline{2G}$ ).

Este circuito puede conectarse de forma que funcione como decodificador de 2 a 4 ó de 3 a 8 ó bien como demultiplexo de 1 a 4 líneas ó de 1 a 8 líneas. Las *figuras 6.8.* y *6.9* muestran los símbolos lógicos y las tablas de verdad del circuito para su funcionamiento como decodificador de 2 a 4 y como

simple demultiplexo de 1 a 4.

Los circuitos demultiplexos pueden usarse para convertir información serie en paralelo. Sin embargo, su uso más frecuente es la síntesis de decodificadores. Cuando las líneas de control representan palabras en binario puro o en BCD, el circuito actúa como decodificador, pasando por ejemplo de binario a decimal. En cada instante sólo estará activa la línea de salida (0, 1, 2, ..., 9) correspondiente al valor binario de la configuración de control (0000, 0001, 0010, ...).

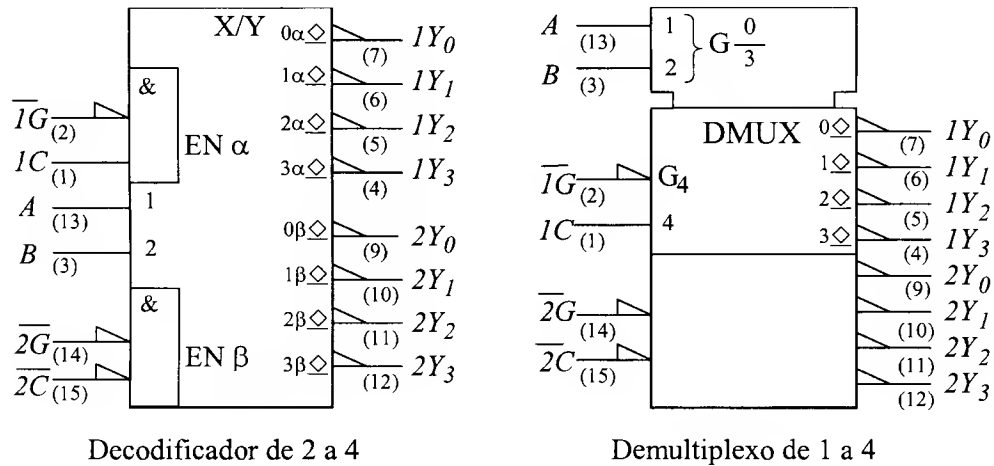


Figura 6.8. Símbolos lógicos del circuito ALS156. (a) Decodificador de 2 a 4. (b) DEMUX dual de 1 a 4.

ENTRADAS				SALIDAS			
Selección	Strobe	Dato					
B	A	$\overline{IG}$	IC	$1Y_0$	$1Y_1$	$1Y_2$	$1Y_3$
x	x	H	x	H	H	H	H
L	L	L	H	L	H	H	H
L	H	L	H	H	L	H	H
H	L	L	H	H	H	L	H
H	H	L	H	H	H	H	L
X	X	X	L	H	H	H	H

ENTRADAS				SALIDAS			
Selección	Strobe	Dato					
B	A	$2\overline{G}$	$2\overline{C}$	$2Y_0$	$2Y_1$	$2Y_2$	$2Y_3$
x	x	H	x	H	H	H	H
L	L	L	L	L	H	H	H
L	H	L	L	H	L	H	H
H	L	L	L	H	H	L	H
H	H	L	L	H	H	H	L
X	X	X	H	H	H	H	H

Figura 6.9. Circuito ALS156. Tablas de verdad para controlar el funcionamiento del circuito como decodificador de 2 a 4 y como DEMUX doble de 1 a 4.



Los términos decodificador y su complementario (codificador) hacen referencia a una función más general de circuitos convertidores de código. Estos circuitos reciben  $n$  líneas de entrada y producen  $m$  líneas de salida pudiendo ser  $n$  mayor ó menor que  $m$ . En general, uno de los extremos es una palabra en binario puro o en decimal codificado en binario (BCD, Binary-Coded-Decimal) y en el otro está la representación decimal o un visualizador de siete segmentos, por ejemplo.

Finalmente, queremos señalar que los circuitos demultiplexos y los decodificadores también pueden usarse, al igual que los multiplexos, para sintetizar funciones lógicas arbitrarias, dado que cualquier circuito decodificador proporciona a su salida los  $2^n$  términos mínimos de las  $n$  variables de entrada. Basta entonces añadir una puerta OR externa para sumar los términos mínimos que posee la función a sintetizar.

Así pues, los circuitos demultiplexores/decodificadores intervienen en el diseño lógico a tres niveles:

- a) *Funciones específicas de distribución de datos.*
- b) *Funciones específicas de decodificación.*
- c) *Como módulos generales de diseño.*

Para las funciones específicas de distribución de datos y de uso más frecuente en decodificación existen soluciones integradas en MSI. La tabla de la *figura 6.10* resume algunas de las soluciones integradas en TTL. Evidentemente, ante un problema de diseño el primer paso deberá ser la revisión de los datos de catálogo buscando la solución o combinación de soluciones específicas que pueda resolvernos el problema de forma inmediata. Cuando tal solución no existe el paso siguiente será diseñar el circuito correspondiente a partir de los módulos disponibles en MSI con la ayuda de algunas puertas lógicas (SSI).

En el apartado anterior hemos visto cómo podían realizarse estos diseños usando multiplexos. Por razones pedagógicas, en este apartado vamos a suponer que sólo disponemos de demultiplexos o decodificadores. En el tema siguiente supondremos que disponemos de memorias PROM y de lógica programable (PLA's y PLD's). Evidentemente, el final de la historia será abordar el problema de diseño suponiendo que disponemos de todo el repertorio actual de funciones combinacionales y secuenciales, programables y no programables, para resolver un problema de diseño.

DESCRIPCIÓN	TIPO de SALIDA	NOMBRE (SN74xxx)
DEMUX de 4 a 16	Tres estados	154
4 a 16	Colector abierto	159
4 a 10 (BCD a Decimal)	Dos estados	42
4 a 10 (Exceso 3 a Decimal)	Dos estados	43
3 a 8 con biestables en las tres líneas de dirección	Dos estados	137
Dual 2 a 4	Dos estados	139
BCD a Decimal	Colector abierto	141
BCD a Siete Segmentos	" "	246

**Figura 6.10.** Algunos ejemplos de circuitos demultiplexos y decodificadores específicos.

Como ejemplos de interés didáctico relacionados con la primera cuestión vamos a estudiar los decodificadores *BCD a decimal* y *BCD a 7 segmentos*.

Un decodificador BCD a decimal necesita cuatro líneas de entrada ( $A, B, C, D$ ), y diez líneas de salida ( $D_0, D_1, \dots, D_9$ ). Ante cada configuración de entrada, sólo está activa una de las líneas de salida, aquella que representa al valor decimal de la configuración binaria de entrada. El código BCD (decimal codificado en binario) más usado es el (8,4,2,1) y coincide con el binario puro hasta llegar al 9 (1001). A partir de aquí, los números de dos o más cifras no se codifican por su valor global en binario, sino por el valor binario de cada una de sus cifras (32, por ejemplo, es 0011,0010). La *figura 6.11*. muestra la tabla de verdad, y las expresiones lógicas de los términos mínimos correspondientes a cada una de las diez salidas ( $D_0, \dots, D_9$ ).

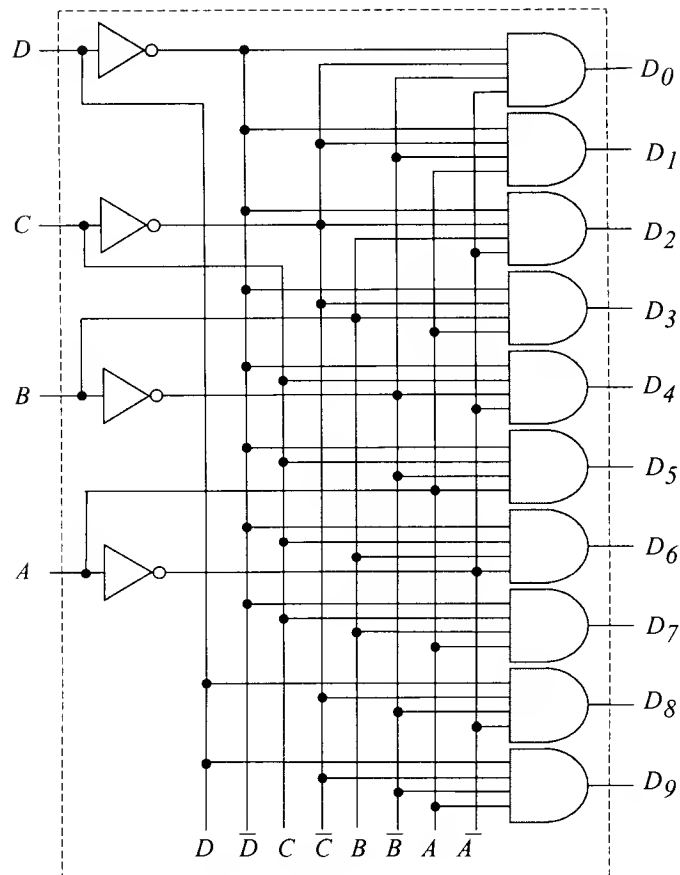
Vamos ahora a comentar la síntesis del decodificador. Como disponemos de cuatro bits, tenemos  $2^4=16$  configuraciones de entrada distintas (0000,...,1111), pero sólo necesitamos 10 (desde 0000=0, hasta 1001=9). Las otras seis pueden usarse para minimizar, como minterms redundantes, que se supone que nunca se va a presentar o bien, para evitar errores, hacer que esas seis configuraciones produzcan una salida característica, por ejemplo poniendo las diez líneas de salida en baja. Aquí hemos elegido esta segunda opción y la *figura 6.12* muestra el circuito correspondiente usando inversores y puertas AND.

CÓDIGO BCD				DECIMAL
8	4	2	1	
$D$	$C$	$B$	$A$	$\{D_i\}$
0	0	0	0	$D_0 = \overline{D} \overline{C} \overline{B} \overline{A}$
0	0	0	1	$D_1 = \overline{D} \overline{C} \overline{B} A$
0	0	1	0	$D_2 = \overline{D} \overline{C} B \overline{A}$
0	0	1	1	$D_3 = \overline{D} \overline{C} B A$
0	1	0	0	$D_4 = \overline{D} C \overline{B} \overline{A}$
0	1	0	1	$D_5 = \overline{D} C \overline{B} A$
0	1	1	0	$D_6 = \overline{D} C B \overline{A}$
0	1	1	1	$D_7 = \overline{D} C B A$
1	0	0	0	$D_8 = D \overline{C} \overline{B} \overline{A}$
1	0	0	1	$D_9 = D \overline{C} \overline{B} A$

**Figura 6.11.** Decodificador BCD a decimal. Tabla de verdad y funciones lógicas asociadas a cada dígito decimal.

Dado que cada salida es un término mínimo, no es necesario realizar ninguna minimización y el funcionamiento del circuito es evidente. Dejamos como ejercicio para el alumno la síntesis del codificador complementario que, partiendo de  $(D_0, \dots, D_9)$  nos genera los cuatro dígitos BCD correspondientes. Es decir,  $(DCBA)$ .

Vamos a comentar ahora el otro ejemplo propuesto que es la decodificación de BCD a "siete segmentos". En el ejemplo anterior sabíamos que una determinada configuración de entrada correspondía por ejemplo al decimal 3, porque se activaba la línea  $D_3$ . En el decodificador con salida siete segmentos se utiliza una estructura rectangular formada por siete barras independientes, tal como se muestra en la *figura 6.13.a*. Cada una de estas barras se realiza electrónicamente mediante un diodo emisor de luz (LED) o mediante tecnología de cristales líquidos. Para excitar estos diodos necesitamos siete funciones lógicas ( $a, b, c, d, e, f$  y  $g$ ) de las cuatro variables de entrada ( $A, B, C, D$ ) de forma que ante cada configuración se activen los diodos necesarios para visualizar el dígito decimal correspondiente. La *figura 6.13.b* muestra la tabla de verdad en la que se incluye el dígito binario, su código BCD y la configuración siete segmentos que lo representa.



**Figura 6.12.** Síntesis del decodificador BCD a decimal.

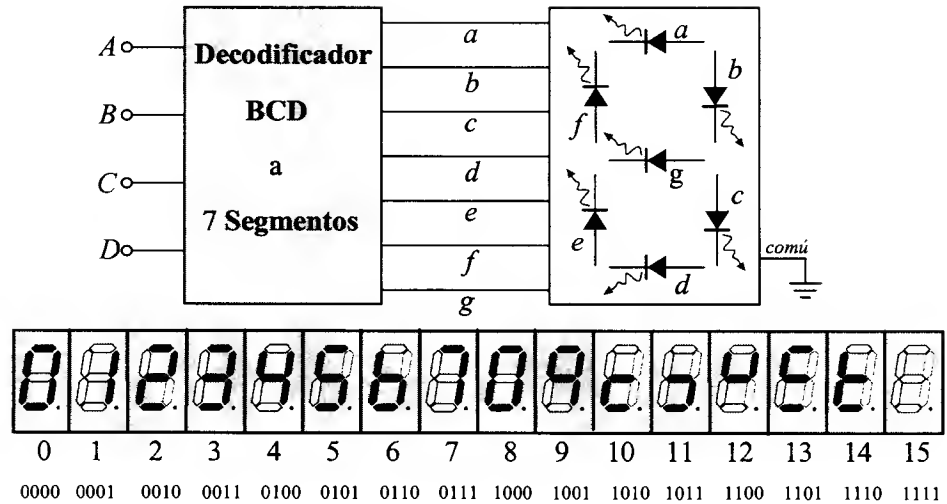
En la *figura 6.14* se muestra el diagrama de Karnaugh correspondiente a la síntesis de la función de activación del segmento "a" y su circuito correspondiente. El resto de las funciones de activación de los otros segmentos (*b, c, ..., g*) se obtienen de forma análoga y se dejan como ejercicio para el alumno. Tal como comentábamos anteriormente, el propósito de este ejemplo es de naturaleza pedagógica ya que existe como circuito integrado (SN7446, 47, 48 y 49).

Para obtener las ecuaciones del segmento "a" observamos el diagrama de Karnaugh de la *figura 6.14.a*. Hemos puesto un uno en los términos mínimos correspondientes pero es más cómodo trabajar con los ceros y obtener así la función complementada ( $\bar{a}$ ). Posteriormente complementamos el resultado y aplicamos los teoremas de DeMorgan. Se obtiene así:

$$\bar{a} = A \bar{B} \bar{C} \bar{D} + \bar{A} C + B D$$

$$\bar{\bar{a}} = a = \bar{A} \bar{B} \bar{C} \bar{D} \cdot \bar{\bar{A}} \bar{\bar{C}} \cdot \bar{\bar{B}} \bar{\bar{D}}$$

[6.8]

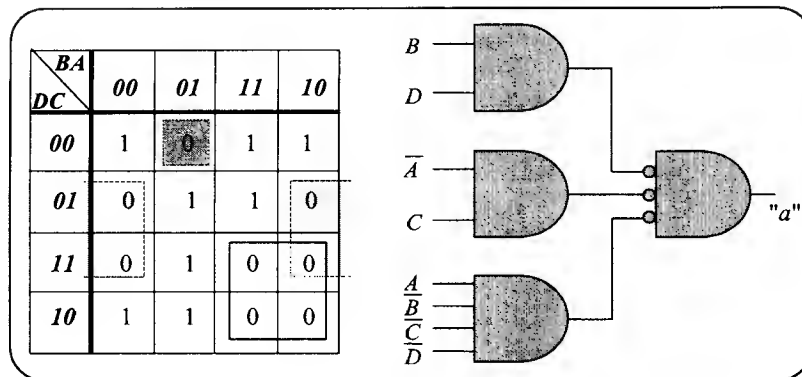


(a)

	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	0	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1
10	1	0	1	0	0	0	0	1	1	0	1
11	1	0	1	1	0	0	1	1	0	0	1
12	1	1	0	0	0	1	0	0	0	1	1
13	1	1	0	1	1	0	0	1	0	1	1
14	1	1	1	0	0	0	0	1	1	1	1
15	1	1	1	1	0	0	0	0	0	0	0

(b)

Figura 6.13. Decodificador BCD a 7 segmentos. (a) Código de diodos usado para representar los dígitos. (b) Tablas de verdad.



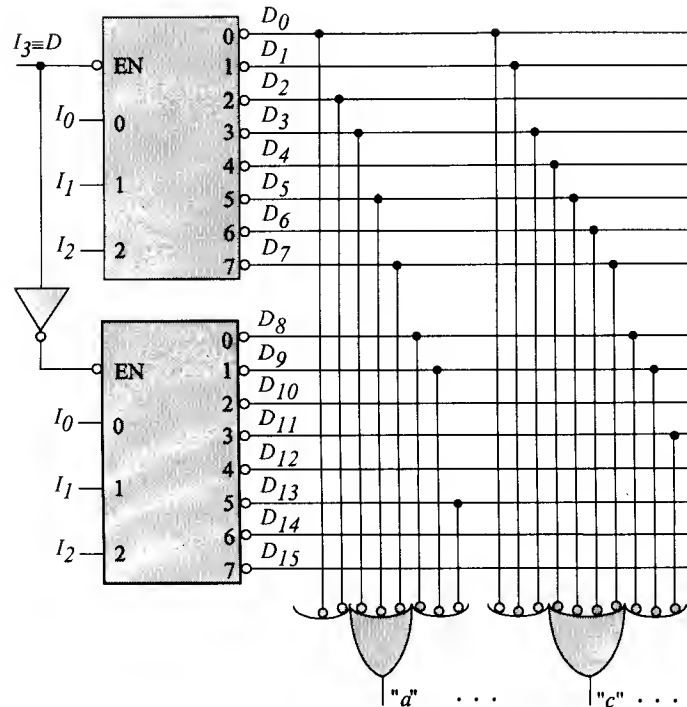
**Figura 6.14.** Síntesis del segmento "a".

Hemos visto la conversión BCD a decimal y BCD a 7 segmentos. De la misma manera podríamos haber visto las conversiones inversas o cualesquiera otras transformaciones entre  $M$  variables de entrada y  $N$  variables de salida.

Vamos a comentar finalmente el uso de decodificadores como módulos universales para la síntesis de funciones lógicas. Recordemos que  $n$  líneas de entrada de datos generan  $2^n$  líneas de salida. Cada línea es un término mínimo, por consiguiente nos basta con añadir tantas puertas "OR" como funciones diferentes deseemos a la salida. Supongamos por ejemplo que disponemos de dos demultiplexos de 3 a 8 tipo SN74138 y que queremos sintetizar las funciones correspondientes a los siete segmentos del decodificador del ejemplo anterior, tal como se muestra en la tabla de la *figura 6.13.b*. ¿Cómo podríamos obtenerlos?

Primero, como necesitamos cuatro variables ( $D, C, B, A$ ) y el demultiplexo del ejemplo sólo posee tres entradas tenemos que usar dos circuitos conectados tal como se muestra en la *figura 6.15*.

Las tres líneas de entrada de datos ( $I_0, I_1, I_2$ ) se usan para las variables ( $A, B, C$ ) y el bit más significativo ( $D$ ) se introduce a través del terminal de facilitación. Cuando está en alta facilita a un decodificador que produce los primeros ocho términos mínimos ( $D_0, \dots, D_7$ ). Cuando está en baja inhibe al anterior decodificador y facilita al siguiente, produciendo los otros términos mínimos ( $D_8, \dots, D_{15}$ ). Tenemos así 16 líneas paralelo con los 16 términos mínimos. Para obtener ahora la función de cada segmento, basta mirar en la tabla de verdad de la *figura 6.13.b* y llevar a una puerta OR los términos correspondientes, sin minimizar. Por razones de claridad en la presentación de la figura solo se muestran los casos correspondientes a los segmentos "a" y "c".



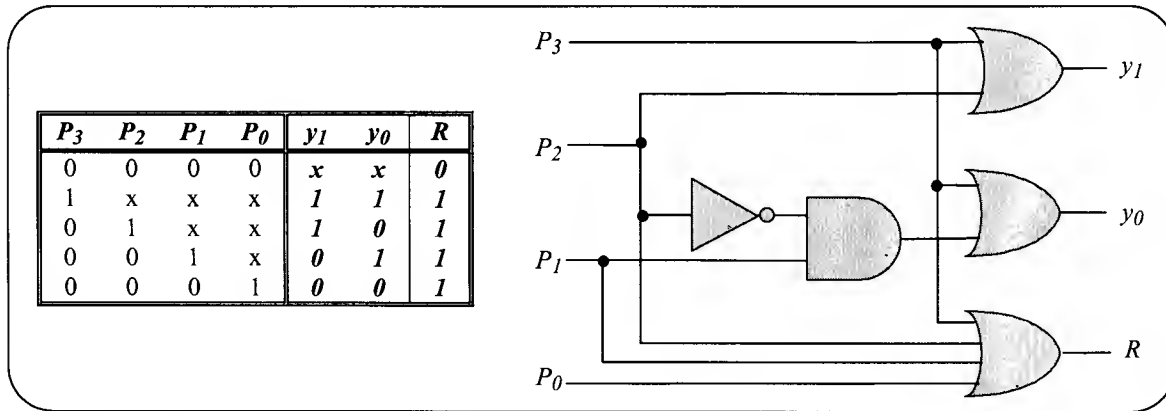
**Figura 6.15.** Uso de circuitos demultiplexores en la síntesis de funciones lógicas. Ejemplo de uso de dos circuitos 3 a 8 conectados a través del terminal de facilitación/inhibición para producir 16 términos mínimos. Las puertas OR con las conexiones indicadas sintetizan los segmentos "a" y "c" del decodificador del ejemplo anterior.

Veremos en el tema siguiente cómo los circuitos conversores de código y la síntesis de funciones arbitrarias a partir de su facilidad para generar los términos mínimos de  $n$  variables son las bases de toda la lógica programable.

### 6.3. CODIFICADORES CON PRIORIDAD

En un circuito codificador se genera a la salida el código correspondiente a la línea de entrada que está activa. Esta función es adecuada siempre que podamos garantizar que en cada momento sólo hay una línea activa. Sin embargo, cuando pueden existir dos o más líneas de entrada activas al mismo tiempo es necesario establecer un criterio de prioridad (un orden en la codificación) de forma que en cada momento sólo se genere el código de la línea que es prioritaria, del conjunto de las que están

activas en ese momento. El circuito que realiza esta función se llama codificador con prioridad. Veamos cómo se puede sintetizar un codificador con prioridad para cuatro líneas de entrada ( $P_3, P_2, P_1, P_0$ ) en los que  $P_3$  es prioritaria a todas las demás. Después es prioritaria  $P_2$ , después  $P_1$  y, finalmente,  $P_0$ . En la *figura 6.16* se muestra la tabla de verdad del codificador y el circuito correspondiente.



**Figura 6.16.** Tabla de verdad y circuito correspondiente al diseño de un codificador con cuatro niveles de prioridad. Se considera que  $P_3, P_2, P_1, P_0$  y  $R$  son activas en baja y que  $y_1$  e  $y_0$  son activas en alta.

Como hay cuatro líneas activas a la entrada nos hacen falta dos líneas de salida para codificar los cuatro niveles de prioridad. Sean estas salidas  $y_1$  e  $y_0$ , y asociemos el grado de prioridad al valor en binario de la configuración que lo codifica. Así, ( $y_1 = 1, y_0 = 1$ ), lo asociamos a  $P_3 = 1$ , independientemente del valor en  $P_2, P_1$ , y  $P_0$ . Análogamente,  $y_1 = 1, y_0 = 0$  lo asociamos a ( $P_3 = 0, P_2 = 1$ ), independientemente del valor de  $P_1$ , y  $P_0$ . Cuando  $P_3 = 0, P_2 = 0$ , y  $P_1 = 1$ , hacemos ( $y_1 = 0, y_0 = 1$ ). Finalmente, cuando está activa  $P_0$ , estando inactivas las otras líneas ( $P_3 = P_2 = P_1 = 0$ ), hacemos  $y_1 = y_0 = 0$ . Hemos añadido una línea adicional ( $R$ ) para diferenciar la situación en la que no hay requerimiento en ninguna de las líneas ( $R = 0$  cuando  $P_3 = P_2 = P_1 = P_0 = 0$ ) de las situaciones en las que al menos una de las líneas está activa.

Por simple inspección de la tabla o por el método más completo de minimización las funciones correspondientes a los términos mínimos que poseen  $y_1, y_0$  y  $R$  es fácil comprobar que:



$$y_1 = P_3 + P_2$$

$$y_0 = P_3 + \overline{P_2} \cdot P_1 \quad [6.9]$$

$$R = P_3 + P_2 + P_1 + P_0$$

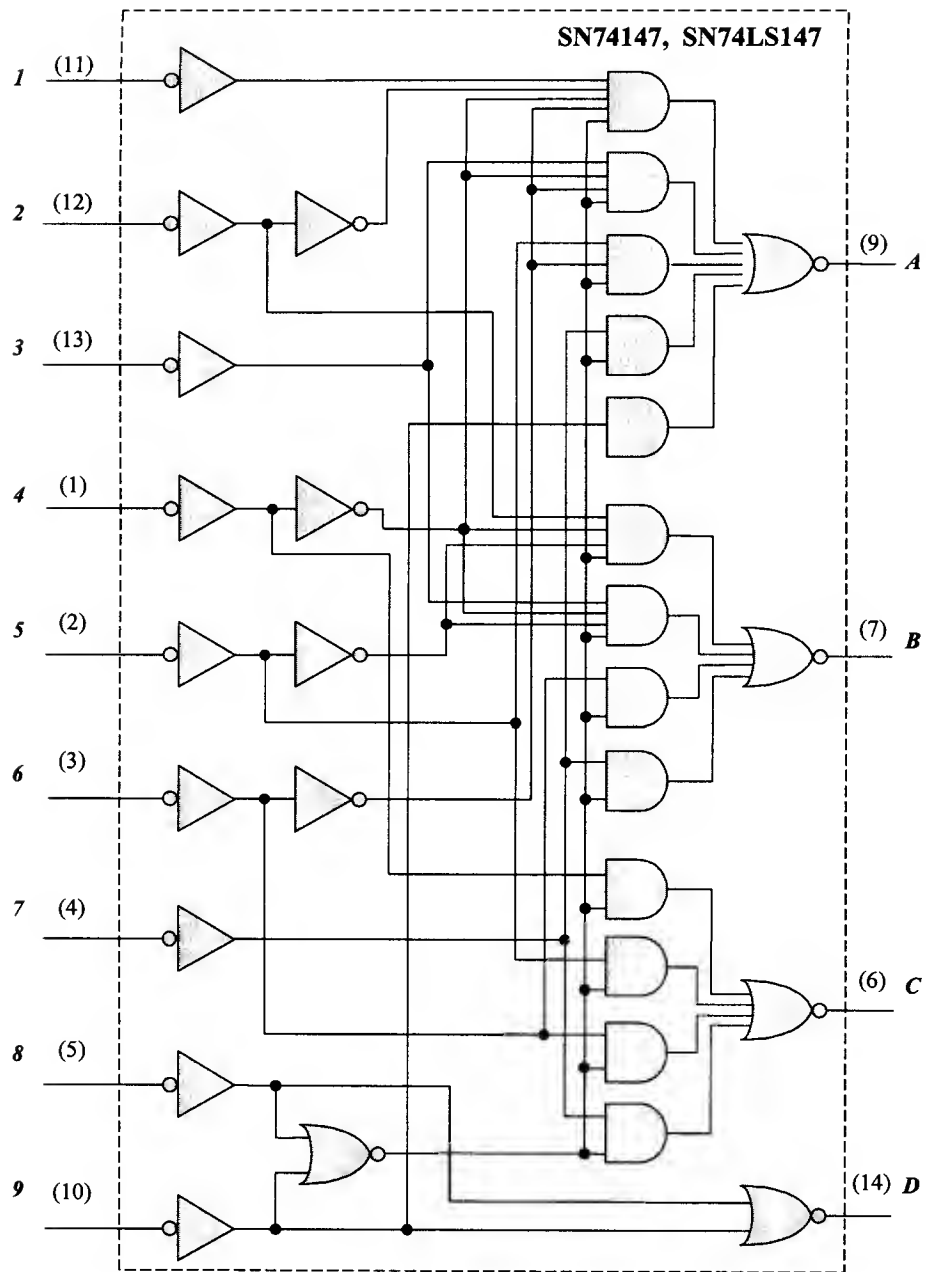
Si suponemos que las entradas y la salida R se consideran *activas en baja*, mientras que  $y_1$ ,  $y_0$  se consideran activas en alta, el circuito resultante es el de la *figura 6.16*.

Hay soluciones integradas en TTL para distintos codificadores con prioridad tales como los circuitos SN74147, 148, 348 y 278. El circuito SN74148 codifica ocho líneas de datos a tres líneas binarias en octal (4-2-1) y permite la conexión en cascada para expandir sin necesidad de otros circuitos externos, usando sólo las líneas *EI* (enable input) y *EO* (enable output). Por otro lado el SN74348 posee las mismas facilidades pero con salidas de tres estados.

El circuito SN74147 codifica 9 líneas de datos en decimal a 4 líneas en BCD (8-4-2-1). La condición implícita del cero en decimal no requiere ninguna condición de entrada ya que el cero se codifica cuando las nueve líneas de datos están en el nivel lógico de alta. Todas las señales de entrada y salida son activas en baja. La *figura 6.17* muestra la tabla de verdad correspondiente a este codificador y la *figura 6.18* muestra su esquema interno.

<i>Entradas</i>									<i>Salidas</i>			
1	2	3	4	5	6	7	8	9	D	C	B	A
H	H	H	H	H	H	H	H	H	H	H	H	H
x	x	x	x	x	x	x	x	L	L	H	H	L
x	x	x	x	x	x	x	L	H	L	H	H	H
x	x	x	x	x	x	L	H	H	H	L	L	L
x	x	x	x	x	L	H	H	H	H	L	L	H
x	x	x	x	L	H	H	H	H	H	L	H	L
x	x	x	L	H	H	H	H	H	H	L	H	H
x	x	L	H	H	H	H	H	H	H	H	L	L
x	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

**Figura 6.17.** Tabla de verdad correspondiente al codificador con prioridad SN74147.



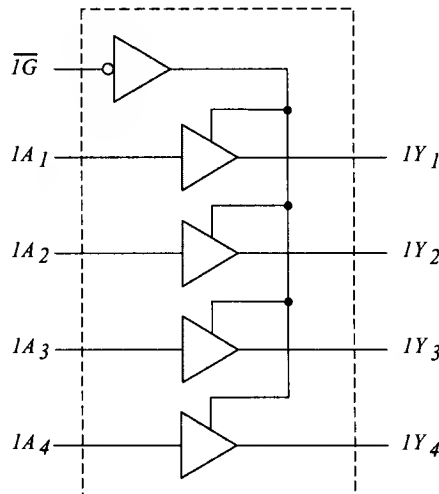
**Figura 6.18.** Circuito codificador con prioridad SN74147 que sólo codifica en BCD la línea decimal de mayor orden

## 6.4. AMPLIFICADORES (BUFFERS-DRIVERS) Y TRANSMISORES-RECEPTORES DE BUS.

Dentro de las funciones de ruta de datos existen un conjunto de circuitos que no realizan cambios lógicos en las señales que procesan pero las preparan para que estén disponibles donde y cuando las necesiten y con el nivel de corriente suficiente para que no existan problemas de impedancia. A su vez, aíslan la fuente de la señal del circuito al que debe ser aplicada. Hay dos tipos de circuitos dentro de este apartado:

- a) *Buffers-drivers.*
- b) *Transmisores-receptores de bus.*

Los primeros son en esencia amplificadores digitales que incrementan los niveles de corriente que se pueden entregar a una carga o absorber de la misma sin confundir el estado. Realizan también funciones de reconstrucción de pulsos en operaciones de lectura/escritura en circuitos de memoria (proporcionando ganancia a las salidas del registro de direcciones). Un ejemplo representativo es el circuito ALS760 con salidas en colector abierto cuyo diagrama se muestra en la *figura 6.19*.



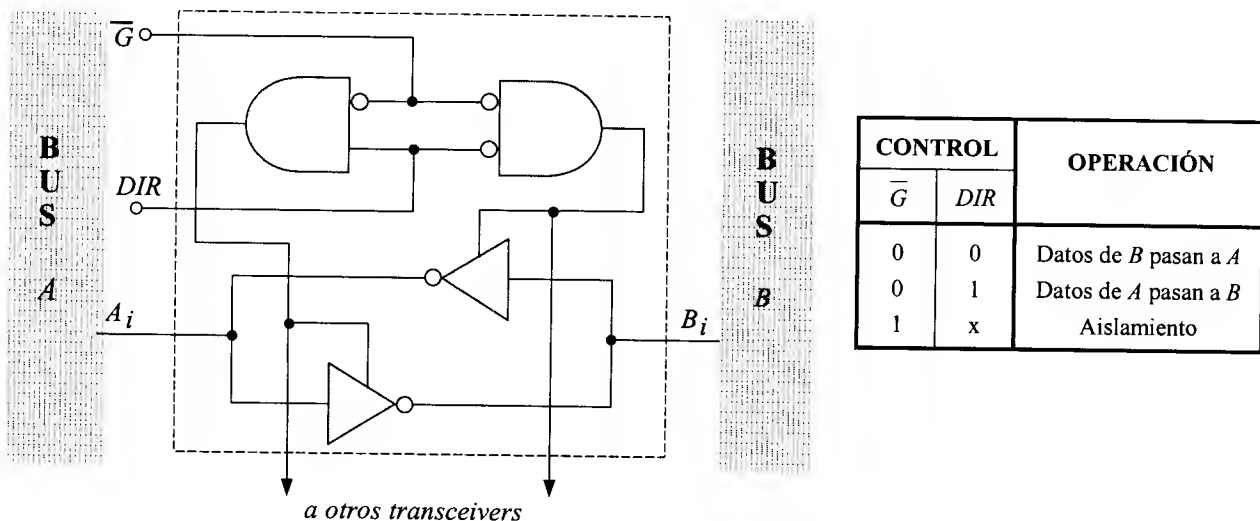
**Figura 6.19.** Circuitos amplificadores (“Buffers-Drivers”) tipo ALS760 (medio circuito).

Estos circuitos amplificadores de corriente están pensados no sólo para mejorar las

características de los circuitos de direccionamiento de memorias con salidas de tres estados sino en todas aquellas aplicaciones en las que la organización de las interconexiones entre los distintos circuitos que constituyen un sistema digital se organizan en buses. Es decir, usando un conjunto de canales conductores comunes a los distintos componentes del sistema digital y soportando tanto la función de transmisión como la de recepción de datos. Electrónicamente se encargan de adaptar impedancias y mantener con precisión los niveles lógicos y sus transiciones durante los intercambios de información con el bus. La salida, en general acepta una configuración en colector abierto o en tres estados.

Los circuitos amplificadores (“drivers-buffers”) son unidireccionales. En cambio los transmisores-receptores de canal (“bus-transceivers”) son bidireccionales. Mantienen para cada dirección de transmisión las características de los “drivers” pero duplican el circuito básico e incluyen dos señales de control,  $DIR$  y  $\overline{G}$ , que facilitan la transmisión en un sentido a la vez que lo inhiben en el sentido contrario. Un ejemplo representativo es la serie ALS640 a 645.

Estos circuitos están pensados para facilitar las comunicaciones asíncronas y bidireccionales entre dos canales de datos,  $A$  y  $B$ , tal como se ilustra en la *figura 6.20*.



**Figura 6.20.** Transmisor-receptor de canal (ALS641).

El sentido de la transferencia lo marca la señal  $DIR$  de forma que cuando está en baja los datos pasan de  $B$  a  $A$  y cuando está en alta pasan de  $A$  a  $B$ . En ambas situaciones la señal de facilitación/inhibición  $\overline{G}$  debe estar en baja. Cuando  $\overline{G}$  pasa a alta las líneas se aíslan del canal.

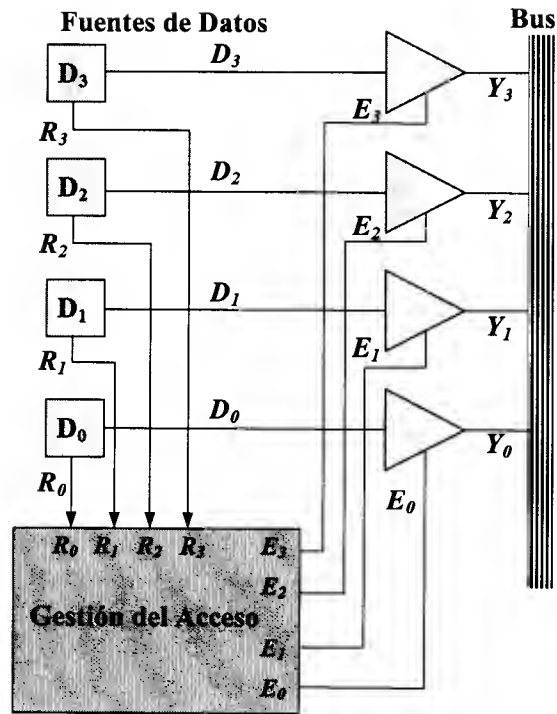
Veamos ahora un ejemplo sencillo de control de las interconexiones con un bus antes de seguir con circuitos integrados transmisores-receptores de bus de mayor grado de complejidad. Supongamos que queremos diseñar un circuito que controle el acceso a un bus de 4 fuentes de señal,  $D_0, D_1, D_2$  y  $D_3$ , de 1 bit cada una suponiendo que:

- I) Cada fuente de señal dispone de un bit adicional de requerimiento de acceso,  $R_i$ , que se pone en alta cada vez que desea acceder al bus.
- II) Cuando dos o más fuentes desean a la vez acceder al bus, se aplica el criterio de prioridad de acceso,  $D_3 > D_2 > D_1 > D_0$ .

Recordemos que los amplificadores de 3 estados (“3-state drivers”), cuya síntesis en TTL y CMOS hemos visto en los temas 3 y 4, proporcionan tres valores a la salida ( $0, 1, Z$ ) donde el valor  $Z$  representa un estado de muy alta impedancia respecto de la fuente y de tierra, de forma tal que a efectos funcionales puede considerarse desconectado del bus. Así, los drivers de 3 estados tienen una línea de datos de entrada,  $D$ , otra línea de salida,  $Y$ , y una tercera línea de control  $E$  (enable line) de forma que cuando esta línea de control está en alta ( $E = 1$ ) la salida,  $Y$ , sigue a la entrada,  $D$  y cuando  $E = 0$ , la salida pasa a alta impedancia (desconecta).

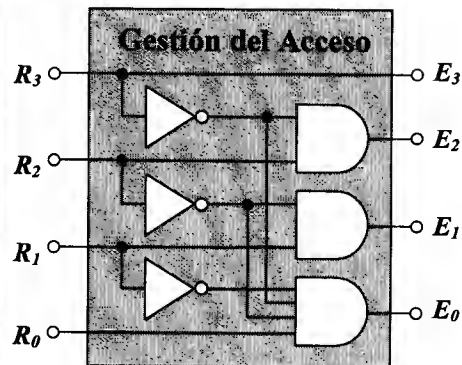
Para el caso de 4 fuentes podemos aceptar el criterio de prioridad  $D_3 > D_2 > D_1 > D_0$ , y proponer el siguiente diseño de la *figura 6.21* que corresponde a las siguientes especificaciones:

- En general, todos los drivers están en alta impedancia ( $E_0=E_1=E_2=E_3=0$ ) con lo que las 4 fuentes están desconectadas del bus.
- Cuando una o varias fuentes desean acceder, pasan a alta su bit de requerimiento de acceso ( $R_i$ ).
- Si sólo hay una  $R_i \neq 0$ , se activa el driver correspondiente,  $E_i$ , y pasa al bus la señal  $Y_i \equiv D_i$ .
- Si hay varias demandas simultáneas, (varias  $R_i \neq 0$ ), el codificador de prioridad selecciona la señal de activación ( $E_j$ ) de la línea prioritaria ( $D_j$ ) y es esta señal la que pasa al bus ( $Y_j \equiv D_j$ ). El resto ( $D_i, i \neq j$ ) quedan bloqueadas en los estados de alta impedancia de sus “drivers”.



Gestión del Acceso							
$R_3$	$R_2$	$R_1$	$R_0$	$E_3$	$E_2$	$E_1$	$E_0$
1	x	x	x	1	0	0	0
0	1	x	x	0	1	0	0
0	0	1	x	0	0	1	0
0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	0

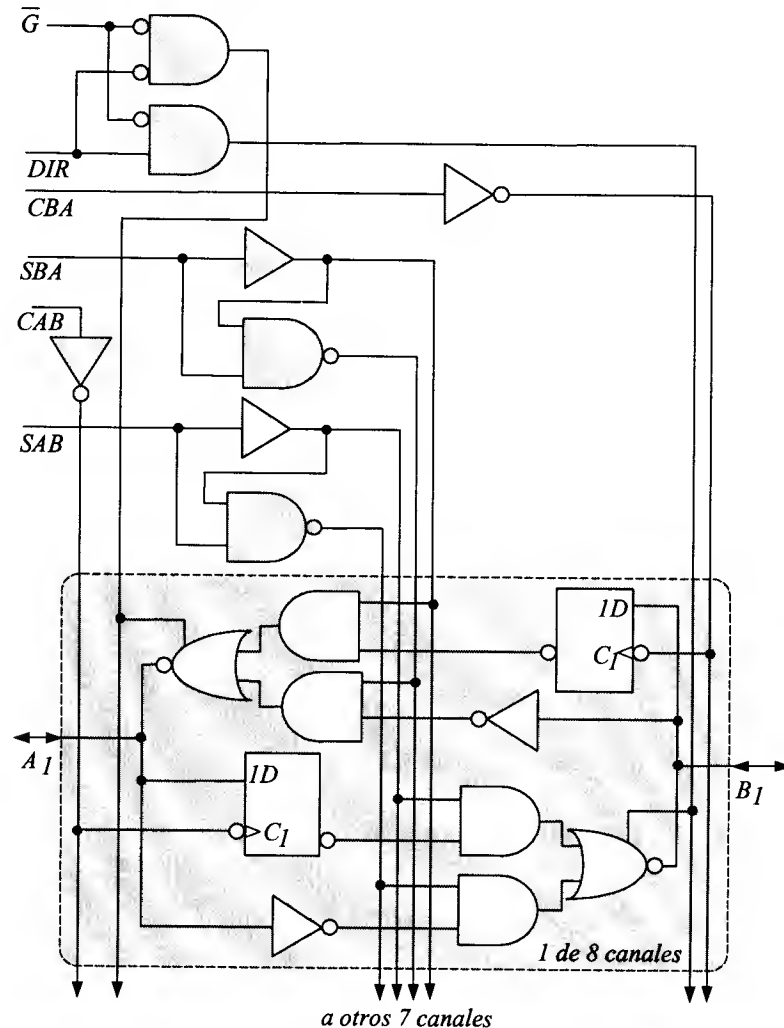
$E_3 = R_3, E_2 = \overline{R_3} R_2, E_1 = \overline{R_3} \overline{R_2} R_1, E_0 = \overline{R_3} \overline{R_2} \overline{R_1} R_0$



**Figura 6.21.** Gestión del acceso a un bus de cuatro fuentes de datos..

El siguiente nivel de complejidad y prestaciones en los circuitos transmisores-receptores de canal incluye elementos de memoria con registros independientes para los datos del canal *A* y los del canal *B* y con posibilidad de multiplexar datos en tiempo real con otros almacenados previamente en

registros. Aunque los biestables y los registros de desplazamiento no se van a estudiar hasta que no lleguemos al tema 8, tras finalizar el estudio de las funciones combinacionales con la introducción de la lógica programable en el tema 7, vamos a comentar aquí por complitud, la serie de “transceivers” ALS646 (3-estados) y ALS347 (colector abierto), aceptando una visión cualitativa del concepto de biestable como circuito que almacena un estado binario y la de registro como un conjunto de biestables. La *figura 6.22* muestra el esquema del circuito ALS646 que es un transmisor-receptor de canal para ocho bits que admite *cuatro modos* de funcionamiento.



**Figura 6.22.** Circuito interno del octal bus transceiver ALS646

La figura 6.23 muestra la tabla de control de los cuatro modos de funcionamiento , que son:

1. Paso directo del canal *B* al canal *A* en tiempo real.
2. " " " " *A* " " *B* " " "
3. Almacenamiento en los registros internos de datos procedentes del *A*, del *B* o de ambos.
4. Transferencia de datos almacenados previamente a los canales *A* ó *B*.

ENTRADAS						DATOS I/O		FUNCIÓN
$\overline{G}$	<i>DIR</i>	<i>CAB</i>	<i>CBA</i>	<i>SAB</i>	<i>SBA</i>	<i>A1-A8</i>	<i>B1-B8</i>	<i>ALS646,647</i>
x	x	↑	x	x	x	Entrada	NE	Almacena <i>A</i> , <i>B</i> , NE
x	x	x	↑	x	x	NE	Entrada	Almacena <i>B</i> , <i>A</i> , NE
H	x	↑	↑	x	x	Entrada	Entrada	Almacena <i>A</i> y <i>B</i>
H	x	H/L	H/L	x	x	"	"	Aislamiento (HS)
L	L	x	x	x	L	Salida	Entrada	Datos bus <i>B</i> al <i>A</i> (TR)
L	L	x	H/L	x	H	"	"	Datos <i>B</i> a bus <i>A</i> (ST)
L	H	x	x	L	x	Entrada	Salida	Datos <i>A</i> a bus <i>B</i> (TR)
L	H	H/L	x	H	x	"	"	Datos <i>A</i> a bus <i>B</i> (ST)

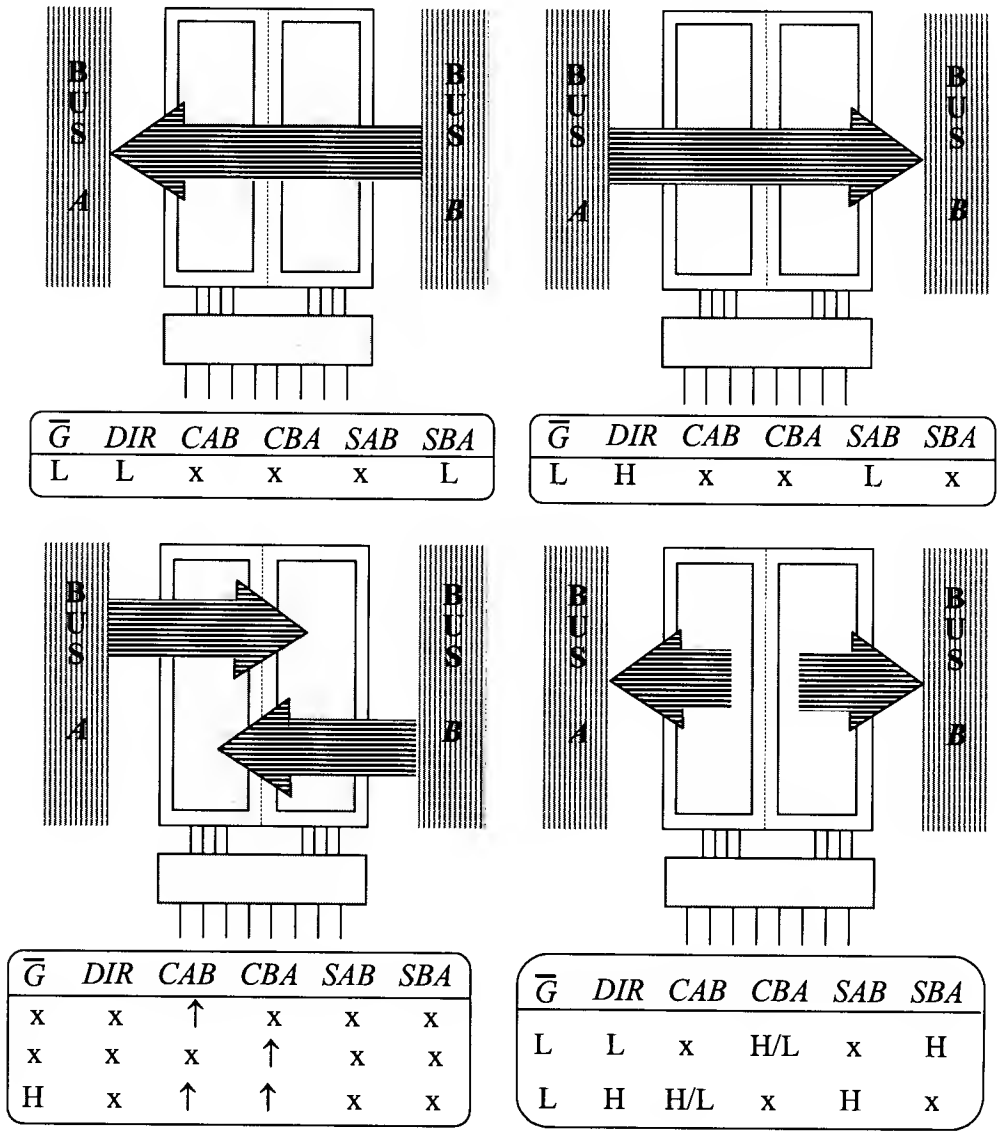
**Figura 6.23.** Tabla de control de modo del ALS 646. Las funciones de salida de dato pueden ser facilitadas o inhibidas por distintas señales en las entradas de control  $\overline{G}$  y *DIR*. Las funciones de entrada de datos están siempre facilitadas, es decir, los datos en los terminales que conectan al bus pueden ser almacenados en cada transición de baja a alta de las entradas de reloj. Las ocho funciones asociadas a los ocho modos específicos de funcionamiento se describen en la parte derecha de la tabla en la que hemos usado el siguiente convenio de notación: *NE* = no especificado. *HS* = retiene el dato almacenado. *TR* = el dato pasa del registro al canal en tiempo real. *ST* = el dato que pasa al canal es el que estaba almacenado en el registro de origen.

Obsérvese que las señales de control ( $\overline{G}$ , *DIR*, *CAB*, *CBA*, *SAB* y *SBA*) no hacen más que multiplexar caminos de transferencia de datos entre un origen y un destino. Así, van abriendo o cerrando las vías que enlazan directamente los dos canales y cada registro interno permitiendo transferencia directa entre canales, transferencia entre canal y registro y, por consiguiente, transferencia directa entre canales a través de registros. La comprensión del funcionamiento del



circuito aconseja considerar a la vez los diagramas de bloques de los cuatro modos fundamentales.

La figura 6.24 los repite de forma anecdótica la operación asociada a cada uno de los cuatro modos, incluyendo también las filas de la tabla de control correspondientes a cada una de esas cuatro transacciones entre los buses A y B.



**Figura 6.24.** Modos de funcionamiento del circuito ALS646.

Existen otros niveles de complejidad en circuitos transmisores-receptores tipo AS865 que son universales y programables. Su estructura interna incluye cada vez más funciones de lógica secuencial por lo que no los vamos a comentar aquí. Conviene recordar finalmente que el propósito pedagógico a lo largo de todo el texto es ilustrar primero la necesidad de una función y los procedimientos de síntesis de sus versiones más elementales y pasar después a conectar con circuitos reales y de uso frecuente, tal como los describen las casas comerciales.

## 6.5. PROBLEMAS (\*)

Hemos visto en este tema la segunda parte de las funciones combinacionales. Las encargadas de controlar el tráfico de datos e instrucciones en un sistema digital. Las dos funciones básicas son la *convergencia* de datos (pasar de  $N$  a 1) y la *divergencia* (pasar de 1 a  $N$ ). En ambos casos hay que abrir y cerrar puertas de forma selectiva y para ello necesitamos generar los términos mínimos de un conjunto de variables de control. Si el mayor número de líneas es  $N$  (estén estas a la entrada o la salida), para su selección necesitamos  $n$  variables de control tales que  $2^n \geq N$ .

Esta necesidad inherente a los circuitos multiplexores y demultiplexores hace que ambos se conviertan en excelentes módulos universales para la síntesis de funciones lógicas arbitrarias.

**E.6.1.** *Diseñar en dos niveles usando MUX de 4 entradas las siguientes funciones lógicas.*

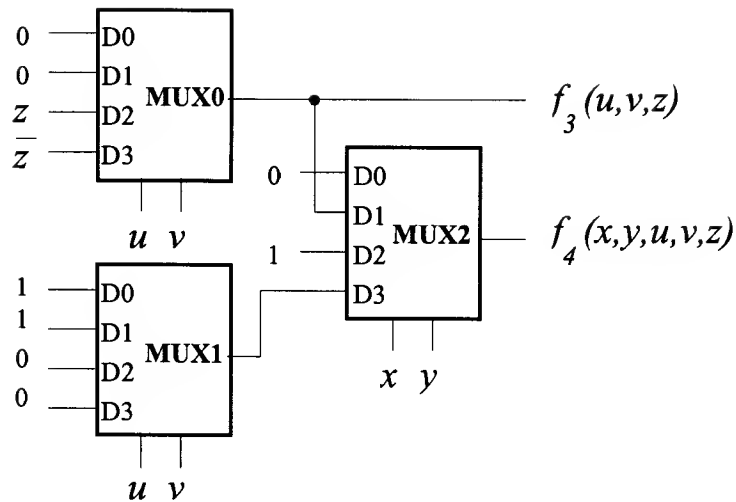
$$f_1(X,Y,Z,U,V) = \overline{XY} + XY\overline{ZU}\overline{V} + \overline{UV}(X+Y+Z)$$

$$f_2(X,Y,Z,U,V) = X\overline{Y}Z\overline{U}\overline{V} + \overline{X}Y\overline{Z}U\overline{V} + X\overline{Z}(U+V)$$

**E.6.2.** *Analizar el circuito de la figura obteniendo los términos mínimos que constituyen las funciones  $f_3$  y  $f_4$ .*

---

(\*) La solución de estos ejercicios se encuentra en el texto de problemas.



**E.6.3.** En la figura 6.10 del desarrollo teórico previo se ha realizado el diseño de un decodificador BCD (código 8421) a decimal. Hay otros códigos como el exceso-3 (obtenido sumando 3 a cada uno de los valores correspondientes) y el código Gray, en el que los números adyacentes sólo difieren en el valor que toma uno de sus bits. La tabla adjunta muestra la representación de los dígitos decimales en cada uno de estos tres códigos binarios.

*Diseñar los circuitos que realizan las siguientes transformaciones:*

- (a) Paso de exceso-3 a Decimal.
- (b) Paso de Gray a Decimal.
- (c) Paso de exceso-3 a Gray.
- (d) Paso de Decimal a BCD (8421).
- (e) Paso de Decimal a Gray.

*En el diseño deberán cumplirse las siguientes limitaciones:*

- 1°. Usar sólo MUX, DEMUX y alguna puerta auxiliar, siempre que sea necesario.
- 2°. Repetir los diseños suponiendo que se dispone no solo de circuitos MUX y DEMUX sino de cuantos circuitos decodificadores existen integrados en MSI. En este segundo caso se permite también el uso de alguna puerta auxiliar, siempre que sea imprescindible.

<b>Dígito Decimal</b>	<b>BCD (8421)</b>	<b>BCD Exceso - 3</b>	<b>Gray</b>
<b>0</b>	0000	0011	0000
<b>1</b>	0001	0100	0001
<b>2</b>	0010	0101	0011
<b>3</b>	0011	0110	0010
<b>4</b>	0100	0111	0110
<b>5</b>	0101	1000	0111
<b>6</b>	0110	1001	0101
<b>7</b>	0111	1010	0100
<b>8</b>	1000	1011	1100
<b>9</b>	1001	1100	1101

**E.6.4.** *En el ejercicio incluido en el desarrollo teórico para ilustrar la conversión BCD a 7 segmentos sólo se ha desarrollado el cálculo correspondiente a los segmentos "a" y "c". Desarrollar aquí los correspondientes a los segmentos "b", "d", "e" y "f" por dos caminos:*

- 1. Usando demultiplexores 3 a 8.*
- 2. Usando puertas NAND, tras minimizar las expresiones correspondientes.*

Usar como punto de partida la tabla de la figura 6.11.



# Preparación de la Evaluación

## Objetivos

- 1: Multiplexos
- 2: Demultiplexos
- 3: Buffers-drivers y Bus-transceivers

## 1. Multiplexos

- 1.1. *Comentar la relación existente entre la obtención de la expresión lógica de la función de salida de un MUX y las formas canónicas de representación de funciones lógicas estudiadas en el tema 1. ¿Qué función adicional incorpora la señal de facilitación/inhibición?.*
- 1.2. *Sintetizar un MUX de 4 a 1 y otro de 8 a 1 usando puertas AND, OR y NOT.*
- 1.3. *¿Cómo podría construir un MUX de cualquier tamaño interconectando varios MUX de 4 a 1 en forma de árbol?. Ilustre el procedimiento diseñando un MUX de 16 a 1 usando sólo como módulo básico el MUX de 4 a 1.*

(Le aconsejamos revisar los problemas E.6.2 y E.6.3. del texto de problemas si encuentra dificultades para contestar a estas cuestiones de evaluación)

- 1.4. *Ya hemos comentado que una de las posibles aplicaciones de los MUX es su uso en la síntesis de funciones combinacionales arbitrarias porque incluyen en su estructura la generación de todos los términos mínimos. Así, basta con seleccionar los necesarios y sumarlos.*

*Veamos si ha comprendido este uso sintetizando las siguientes funciones:*

$$f_1 = \bar{x}y + \bar{u}\bar{v}xyz + u\bar{v}(x + y + z)$$

$$f_1 = xyz\bar{u}\bar{v} + \bar{x}y\bar{z}uv + \bar{x}z(u + v)$$

- a) *Con circuitos MUX de 4 a 1.*

b) Con circuitos MUX de 8 a 1.

## 2. Demultiplexos

2.1. ¿Qué elementos de un multiplexo comparte un Demultiplexo?

2.2. Construir un DEMUX de 1 a 4 con AND, OR y NOT

2.3. Explicar la idea de árbol demultiplexor. ¿Cómo podría diseñar un demultiplexo de 1 a 32 a partir de un demultiplexo de 1 a 4 y cuatro demultiplexos de 1 a 8?. Intente describir de forma algorítmica el procedimiento general de diseño.

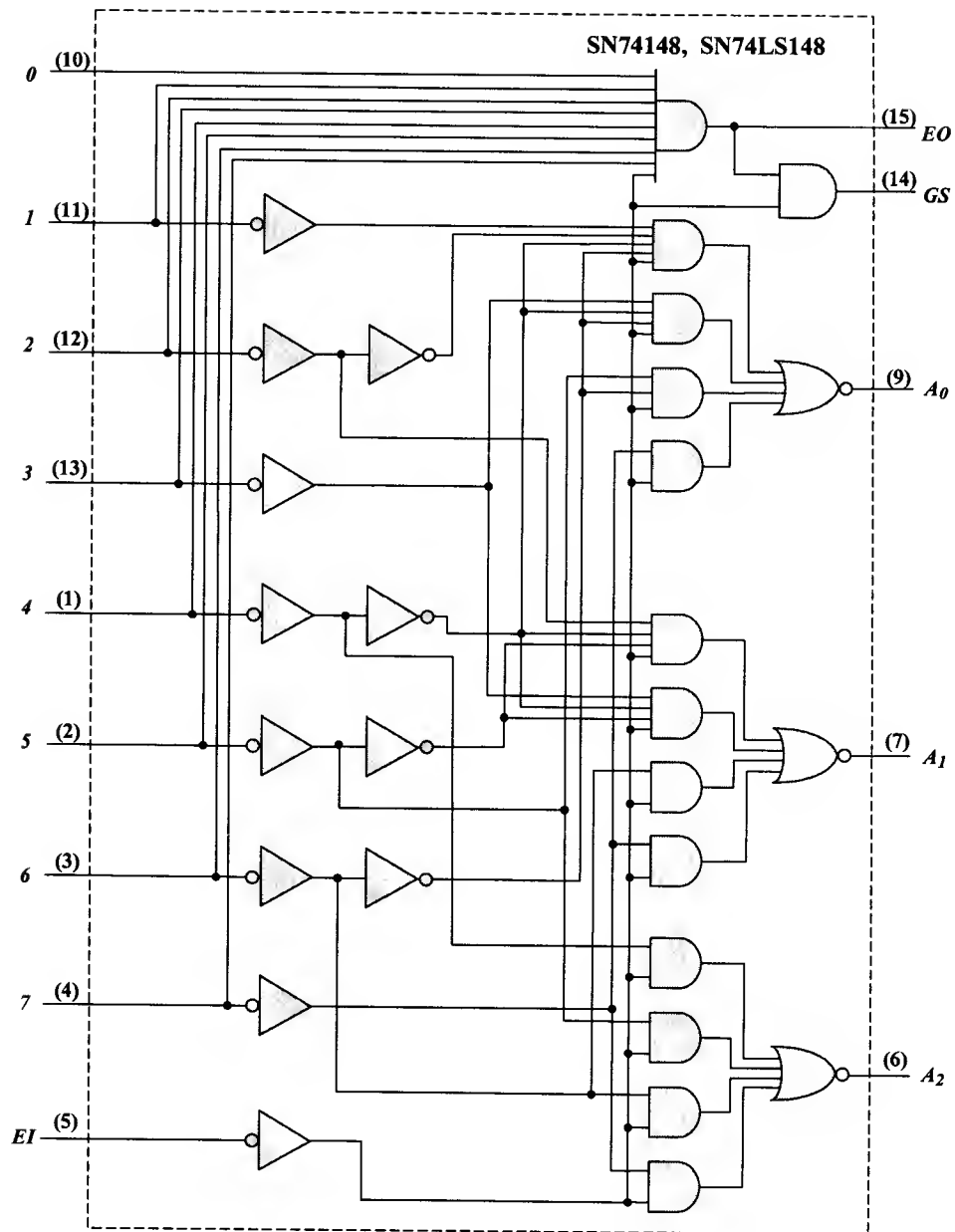
2.4. Como recordará del estudio de este tema: “Los términos decodificador y su complementario (codificador) hacen referencia a una función más general de circuitos convertidores de código. Estos circuitos reciben  $n$  líneas de entrada y producen  $m$  líneas de salida pudiendo ser  $n$  mayor o menor que  $m$ . En general, uno de los extremos es una palabra en binario puro o en decimal codificado en binario (BCD, Binary-Coded-Decimal) y en el otro está la representación decimal o un visualizador de siete segmentos, por ejemplo”.

En el problema E.6.4 se desarrollan un conjunto de ejemplos. ¿Podría integrarlos proponiendo un circuito que, dependiendo del valor de una variable de control  $C$ , actuara como convertidor exceso-3 a decimal ( $C=1$ ) ó Gray a decimal ( $C=0$ )

2.5. En el ejercicio E.6.5. del texto de problemas se desarrollan los cálculos para la conversión BCD a 7-segmentos usando DEMUX de 3 a 8 y en el caso concreto de los segmentos “a” y “g”. ¿Podría obtener ahora los correspondientes a los segmentos “b” y “c”?

2.6. Además de los circuitos SN74147 hay otras opciones de circuitos codificadores de prioridad tales como la SN74148 y SN74348. La primera codifica ocho líneas de datos a tres líneas binarias (4-2-1) octal) y permite la conexión en cascada para expandir sin necesidad de otros circuitos externos, usando sólo las líneas EI (enable input) y EO (enable output). Por otro lado el SN74348 posee las mismas facilidades pero con salidas de tres estados.

El circuito de la figura corresponde al SN74148 y se usa en procesos de conversión de código y generadores. Trabaja en lógica negativa (líneas activas en baja). Analice el circuito comprobando que efectivamente genera código binario octal. (Obtener las expresiones de  $A_0$ ,  $A_1$ ,  $A_2$ ).



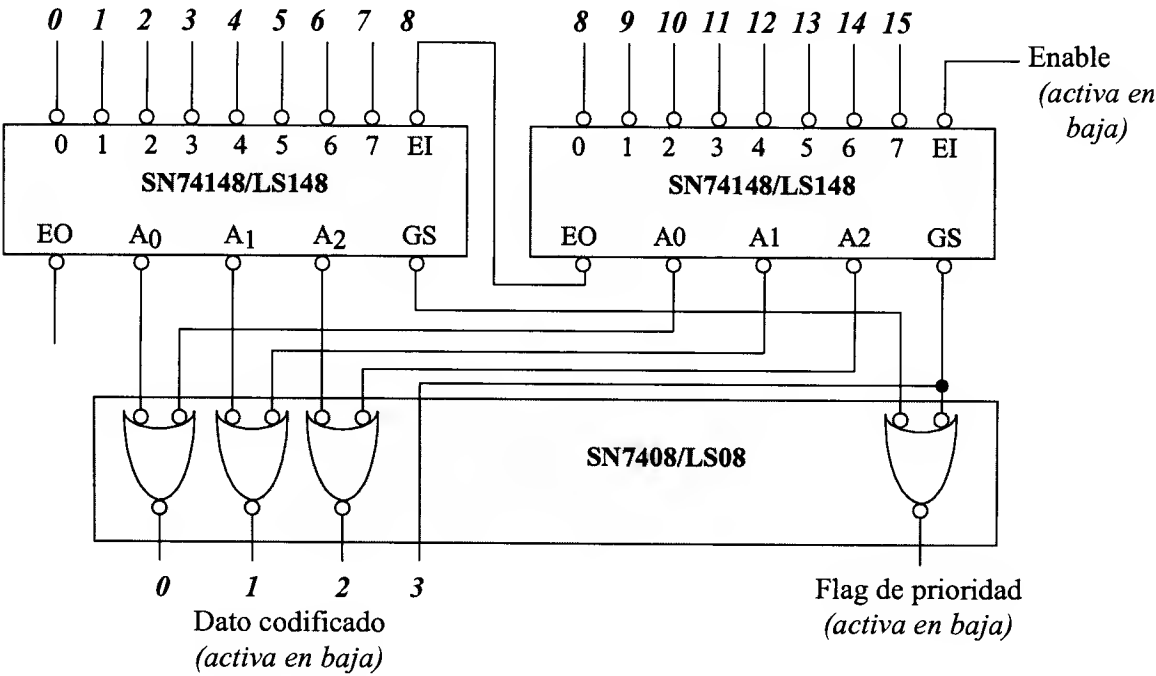
Obtenga también las expresiones lógicas de las líneas EO y GS (bit de prioridad).

¿Qué función realizan las líneas de facilitación de entrada (enable input, EI) y facilitación de salida (enable output, EO) en la conexión de varios de estos circuitos para permitir la

expansión octal? Para facilitar el trabajo incluimos la tabla de verdad y el esquema de conexión para la expansión.

Tabla funcional

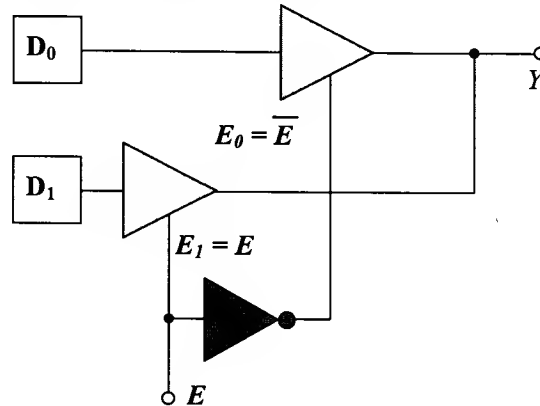
Entradas									Salidas				
E1	0	1	2	3	4	5	6	7	A2	A1	A0	GS	E0
H	x	x	x	x	x	x	x	x	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	x	x	x	x	x	x	x	L	L	L	L	L	H
L	x	x	x	x	x	x	L	H	L	L	H	L	H
L	x	x	x	x	x	L	H	H	L	H	L	L	H
L	x	x	x	x	L	H	H	H	L	H	H	L	H
L	x	x	x	L	H	H	H	H	H	L	L	L	H
L	x	x	L	H	H	H	H	H	H	L	H	L	H
L	x	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H





### 3. Buffers-Drivers y Bus-Transceiver

- 3.1. Enumere las funciones básicas de los amplificadores digitales y de los transmisores-receptores de bus.
- 3.2. Sabemos cómo funcionan los drivers de tres estados para lógica positiva. ¿Podría pasar el sencillo ejemplo de la figura a lógica negativa?



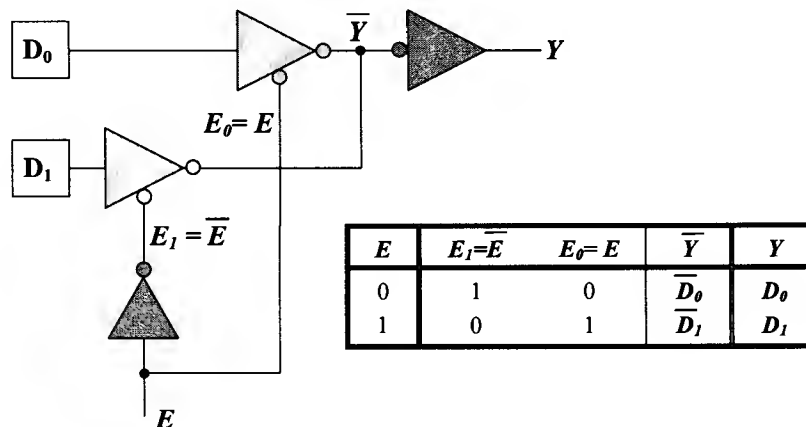
#### Solución

Para el circuito del enunciado y teniendo en cuenta que trabaja en lógica positiva la tabla de verdad es:

$E$	$E_1=E$	$E_0=\overline{E}$	$Y$
0	0	1	$D_0$
1	1	0	$D_1$

Ahora tenemos que diseñar un circuito que realice la misma función, pero que trabaje con lógica negativa. Es decir, siendo activas las líneas de facilitación ( $E_1$ ,  $E_0$ ) cuando están en baja. Así, cuando  $E=0$ , como la salida debe ser  $D_0$ , el buffer que debe dejar pasar la señal es el que controla  $E_0$ , es decir  $E_0=0$  ya que estamos trabajando con lógica negativa. Como la salida se debe de interpretar también en lógica negativa, a la salida tendremos el complemento de la entrada del buffer que controla  $E_0$ ,  $\overline{D_0}$ . Además, el buffer que controla  $E_1$ , debe presentar alta impedancia y quedar desconectado. En cambio, cuando  $E=1$ , el buffer controlado por  $E_0$  deberá presentar alta impedancia (desconectado) y ser ahora la señal controlada por  $E_1$  la que pase a la salida,  $\overline{D_1}$ .

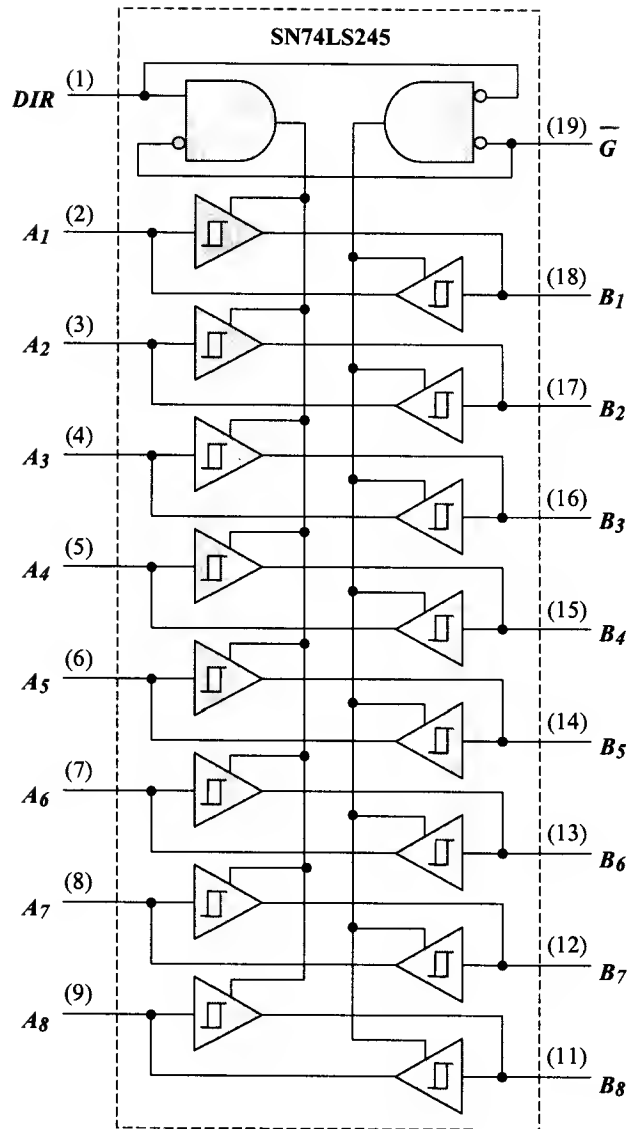
Como el drivers opera con lógica negativa, la orden de facilitación se pasa por un inversor antes de entrar al buffer. Además, si no deseamos la inversión que realiza el driver con la entrada, podemos añadir un inversor. Así, el circuito y la tabla de verdad es:



3.3. El esquema de la figura corresponde a un transmisor receptor de bus octal bidireccional con salidas de tres estados. Está pensado para gestionar la transmisión bidireccional asíncrona entre buses de datos. El circuito facilita la transmisión del bus A al bus B o del bus B al bus A dependiendo del valor lógico de la variable de control (DIR). La entrada de facilitación ( $\overline{G}$ ) se puede usar para inhibir el circuito de forma que los buses A y B queden efectivamente aislados, de acuerdo con la siguiente tabla:

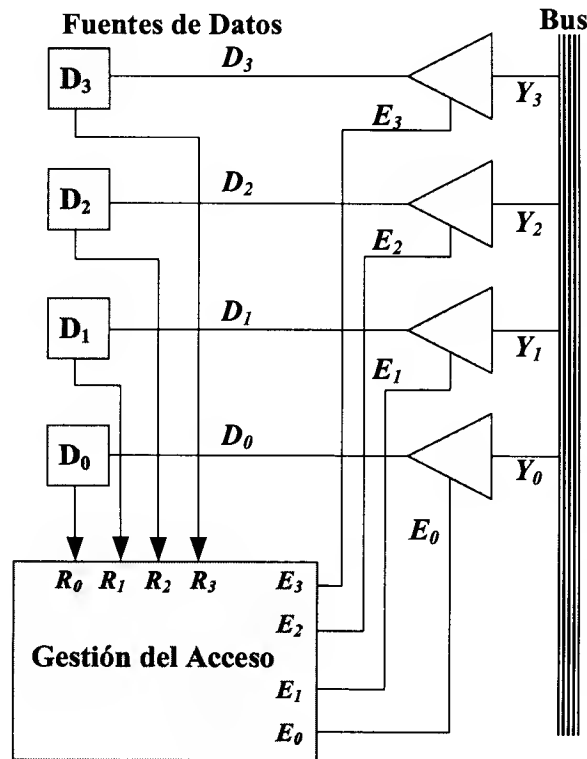
Facilitación	Control de la dirección de transmisión	Operación
L	L	Datos pasan de B a A
L	H	Datos pasan de A a B
H	x	Aislamiento

- Explique su funcionamiento y compruebe la tabla de verdad.
- Suponga, para facilitar el dibujo, que sólo nos quedamos con el control y con un bit. ¿Cómo podría usar este módulo para resolver los problemas de gestión de comunicación vistos en los ejercicios incluidos en el tema?.



3.4. Hemos visto en los ejercicios incluidos en el tema el diseño de un circuito para el control del acceso a un bus de cuatro fuentes de señal  $D_0$ ,  $D_1$ ,  $D_2$  y  $D_3$ . ¿Podría resolver ahora el problema complementario. Es decir, diseñar un circuito para la gestión de toma de datos de un bus por cuatro receptores  $D_3$ ,  $D_2$ ,  $D_1$ ,  $D_0$  con prioridad ( $D_3 > D_2 > D_1 > D_0$ ) de forma que cuando un receptor quiere tomar el dato del bus le envía una señal de requerimiento ( $R_3$ ,  $R_2$ ,  $R_1$ ,  $R_0$ ) que sirve para abrir el camino de sus drivers de tres estados. Si más de un receptor quiere tomar

datos del bus en un mismo intervalo de tiempo se usa el criterio de prioridad mencionado. Le sugerimos un esquema inicial como el de la figura adjunta



3.5. ¿Podría integrar las dos tareas de enviar y recibir en un único circuito gestor de las comunicaciones con un bus de tal manera que cada uno de los dispositivos  $D_3$ ,  $D_2$ ,  $D_1$ ,  $D_0$ , pudiera actuar como emisor, enviando su señal al bus y receptor, tomando la señal del bus?. Considere la posibilidad de usar dos tipos de señales complementarias:

- $R_3$ ,  $R_2$ ,  $R_1$ ,  $R_0$  para cuando desea enviar señal.
- $S_3$ ,  $S_2$ ,  $S_1$ ,  $S_0$  para cuando desea recibir señal.

Le sugerimos reflexionar sobre el circuito del tipo SN74245 que hemos incluido en el ejercicio de preparación de la evaluación 3.3.



Ya hemos comentado en la componente tutorial de otros *temas* que las soluciones que aquí se estudian (tecnología integrada en media escala) son en la actualidad componentes reutilizables de una librería de motivos a partir de la cual se diseñan otros circuitos de mayor complejidad. Todo el contenido de los temas 5 y 6 tiene, esencialmente, propósitos pedagógicos y es muy aconsejable que el alumno entre a navegar en los sitios de la red de las distintas casas comerciales para conseguir un conocimiento más completo y práctico del repertorio real que se ofrece en la actualidad en las distintas tecnologías para el diseño de funciones aritmético-lógicas y de ruta de datos (<http://www.fairchildsemi.com>, [.ibm.com](http://www.ibm.com), [st.com](http://www.st.com), [onsemi.com](http://www.onsemi.com))

Los alumnos que estén cursando a la vez asignaturas sobre Estructura y Tecnología de Computadores tienen una magnífica ocasión para hacer más comprehensivo su conocimiento. Aquí han visto la orientación hacia circuitos y allí ven la orientación hacia la integración de los circuitos en una estructura global más amplia. Aquí hablamos de transferir de forma asíncrona señales del bus *A* al *B* o viceversa. Allí estudiarán la organización por buses y verán cómo estos circuitos (drivers y transceivers) se usan, por ejemplo, para enlazar buses de datos, para conectar con los registros de datos y/o direcciones de memoria, etc. Es decir, el uso específico, integrado en un plan global, de todos los circuitos de operaciones aritmético-lógicas y de tráfico controlado de datos, direcciones e instrucciones. Aquí no nos preocupa que un conjunto de señales binarias vayan a ser interpretadas más tarde como datos, direcciones, instrucciones o señales de control en un procesador. Para nosotros son señales lógicas con dos valores de tensión, que se combinan de acuerdo con las leyes propias del Algebra de Boole para generar otras señales de la misma naturaleza.

## REFERENCIAS BIBLIOGRÁFICAS

1. Breeding, K.J.: "*Digital Systems Fundamentals*". Prentice Hall, 1992.
2. Texas Instruments. "*The TTL Data Book*". 1987.



# TEMA 7

---

## ***Lógica Combinacional Programable***

---

### **+ CONTEXTO**

*En los temas 5 y 6 hemos visto cómo se pueden diseñar circuitos lógicos (sumadores, ALUs, MUX, DEMUX, etc.) de forma modular y estructurada a partir de puertas AND, OR y NOT en dos niveles. Partíamos del valor de un conjunto de variables y sus negadas, obteníamos los productos necesarios y después los sumábamos para obtener las funciones de salida. Tanto la función de los módulos de partida como el esquema de conexiones estaban totalmente especificados.*

*Este procedimiento es recursivo. Es decir, si ya tenemos diseñados y disponibles en tecnología integrada esos módulos intermedios (integración en media escala, MSI), ahora podemos usarlos para diseñar funciones lógicas de mayor complejidad (usando multiplexos para conectar ALUs con circuitos de memoria y/o buses, etc...) pero, de nuevo, la función de esos módulos más complejos (integración en gran escala, LSI y muy gran escala, VLSI) está totalmente especificada. Así ha evolucionado esta línea de diseño hasta conseguir integrar un gran número de funciones en un único circuito estándar de gran complejidad y uso masivo, como por ejemplo un microprocesador, pero con muy poca o ninguna posibilidad de cambiar su estructura física (la conectividad de sus circuitos) de*

*acuerdo con las exigencias del usuario. La clave de esta línea de diseño es el carácter absolutamente general del circuito desarrollado (el microprocesador), lo que permite su fabricación en gran escala y a muy bajo coste. Cualquier cambio de aplicación debe realizarse a través de la programación convencional, en general en un lenguaje de alto nivel, y lejos de la electrónica digital.*

*Existe otra línea asociada a necesidades de diseño que por flexibilidad, especificidad y propiedad del producto busca soluciones de "propósito especial". Para estos casos las soluciones son 1) de diseño completo ó 2) de diseño parcial, sobre estructuras flexibles. El diseño completo ("custom") ofrece considerables ventajas en términos de tamaño y precio de las unidades finales pero posee las desventajas asociadas al tiempo y al presupuesto del desarrollo. Se consigue disminuir el coste estandarizando los procesos de desarrollo y producción. Obtenemos así circuitos a la medida de las necesidades del cliente pero respetando ciertas limitaciones ("semi-custom") en lo referente a los módulos de partida ("arrays de puertas" o "células standard") y al propio proceso de fabricación.*

*Finalmente, existen aplicaciones en las que la confidencia del diseño y el tiempo de desarrollo son variables esenciales. Cuando la velocidad de paso al mercado es esencial, la solución es el diseño sobre estructuras flexibles. El campo que cubren es el de "circuitos integrados de aplicaciones específicas" (ASIC -Application Specific Integrated Circuits) y la tecnología es la de redes de puertas programables FPGA, (Field Programmable Gate Array), con memorias RAM estáticas para programar las conexiones.*

*Estos circuitos de lógica programable de última generación combinan los beneficios del diseño VLSI completo, a la demanda del usuario, con las ventajas de diseño, producción y velocidad de paso al mercado de los productos estándar. Los diseñadores definen las funciones lógicas que desean, las realizan en lógica programable, las evalúan en su entorno real y pueden refinar el diseño en horas, frente a las semanas necesarias en las soluciones "custom" y semi-custom".*

*Para las funciones de complejidad equivalente a varios circuitos MSI, la solución más eficiente es la PLD (Dispositivos Lógicos Programables) basados en matrices de circuitos AND, seguidas de matrices de circuitos OR, organizadas en una arquitectura plana y con conectividad programable. Su estudio es el objetivo de este tema. Las soluciones CPLD y FPGA más complejas, que incluyen funciones secuenciales, mucha mayor densidad de integración, memorias empotradas, etc., serán estudiadas posteriormente (tema 13)*

*Conviene observar cómo ambas filosofías de diseño se complementan y se mezclan. En efecto, en el desarrollo de los circuitos lógicos programables se introducen constantemente nuevas posibilidades de programación de forma que también se tiende hacia unos pocos tipos de circuitos básicos genéricos y programables, donde un mismo circuito puede realizar una amplia variedad de funciones dando lugar a una gran diversidad de implementaciones específicas (personalizadas) y competitivas en términos de tiempo de desarrollo, consumo, tamaño, forma, confiabilidad del diseño y capacidad de reprogramación. La única restricción importante es la limitada capacidad computacional de los módulos de partida. Sin embargo, la rápida evolución de la tecnología hace prever soluciones híbridas cada vez más potentes, con PLDs que incluyen memorias y módulos de proceso para reprogramar sus conexiones, y con potentes entornos software para facilitar el paso desde las especificaciones funcionales hasta el circuito final que las implementa.*

## **++ CONOCIMIENTO PREVIO NECESARIO**

*Vamos a estudiar la lógica programable en dos temas:*

**Tema 7: *Lógica Combinacional Programable (PLDs)***

**Tema 13: *Lógica Secuencial Programable (CPLDs y FPGAs)***

*La razón de distinguir estos dos contenidos es clara. Para estudiar los PLDs sólo necesitamos lo que ya hemos visto en los temas previos. En cambio, para estudiar la Lógica Secuencial Programable (los CPLDs, Complex Programmable Logic Devices y los FPGAs, Field Programmable Gate Arrays) necesitamos conocer los biestables, los circuitos secuenciales básicos, (contadores, registros) las memorias RAM y EEPROM, etc. Por eso la dejamos para más tarde. Así pues, los conocimientos previos necesarios para el tema 7 se encuentran en los temas 1, 5 y 6, que ya hemos visto con la salvedad del uso cualitativo del concepto de biestable, necesario para describir las macroceldas.*

## **+++ OBJETIVOS DEL TEMA**

*Dentro de la lógica combinacional programable, y bajo el nombre genérico de PLDs, existen distintas arquitecturas:*

- PROM (Programmable Read Only Memory)



- PAL (Programable Array Logic)
- PLA (Programable Logic Array).

*Las diferencias se encuentran en el grado de programabilidad de cada una de ellas y en la parte de la función en la que se asienta esa programabilidad.*

*El objetivo general de este tema es comprender la alternativa de diseño lógico que suponen las PLDs. Sus objetivos más concretos son los siguientes:*

**Objetivo 1:** *Conocer la estructura interna de las memorias de sólo lectura (ROM, PROM, EPROM, EEPROM y FLASH). Saber cómo pueden usarse para distintos tipos de aplicaciones.*

**Objetivo 2:** *Conocer las arquitecturas PAL y PLA y saberlas usar en el diseño en lógica combinacional.*

## ++++ GUÍA DE ESTUDIO

*Al contrario que en los dos temas anteriores, el conocimiento propio de este tema se encuentra muy disperso y no hay mucho acuerdo entre los distintos fabricantes sobre nomenclatura. Además, el campo de las PLDs es muy activo y está en fase de transformación continua. Por eso, en general, no entraremos en las ofertas específicas de los distintos fabricantes, sino que haremos énfasis en los aspectos conceptuales básicos, compartidos por toda la tecnología de las PLDs, con la esperanza de que en la posterior actividad profesional del alumno pueda comprender los datos de catálogo y las innovaciones que, con seguridad, se irán introduciendo.*

*Hay un problema en el caso de los PLDs más avanzados y es que incluyen en su estructura interna pequeños procesadores y registros y otros circuitos que son propios de la lógica secuencial y, por consiguiente, de acuerdo con el plan general de este texto no deberían presentarse hasta después de los temas 9 y 10. Por eso, aquí se hará una referencia muy superficial e intuitiva a estas funciones.*



---

# Contenido del Tema

---

## 7.1. PROCESAMIENTO DIGITAL DE LA INFORMACIÓN

La lógica programable busca una estructura modular y general que nos permita obtener cualquier función lógica a través de la programación eléctrica o funcional del esquema de conexión de esos módulos. El modelo formal subyacente a todos los PLDs (programmable logic devices) procede de la representación en forma normal disyuntiva de la función múltiple que enlaza  $n$  variables de entrada con  $m$  variables de salida. Así, si tenemos los términos mínimos de  $n$  variables de entrada, podremos obtener en paralelo cualquiera de las  $m$  funciones de salida con tal que seamos capaces de “programar” a nivel de circuitos las matrices de conexión adecuadas.

Veamos este origen de las PLDs pensando sobre un modelo sencillo de dos funciones ( $y_0, y_1$ ) de dos variables ( $x_0, x_1$ ).

Recordemos que la forma normal disyuntiva representa todas las funciones lógicas como sumas de productos. Así, para dos variables ( $x_1, x_0$ ), tenemos cuatro términos mínimos ( $x_1x_0, x_1\bar{x}_0, \bar{x}_1x_0, \bar{x}_1\bar{x}_0$ ) y cada una de las 16 posibles funciones de dos variables ( $f_0, f_1, f_2, \dots, f_{15}$ ) se obtiene sumando los términos mínimos correspondientes a las coordenadas en alta del vector lógico que codifica el subíndice de la función:

$$f_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad f_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad f_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad f_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \quad \dots, \quad f_{15} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

[7.1]

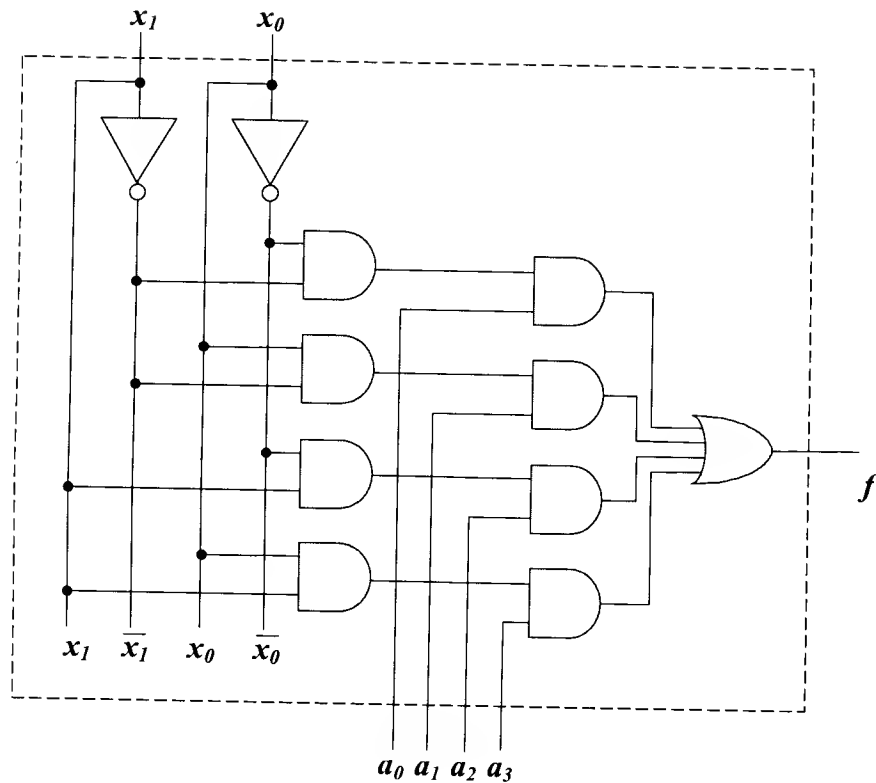
Evidentemente, con 2 variables tenemos 4 términos mínimos y con 4 términos mínimos podemos obtener 16 funciones lógicas diferentes que son las que se muestran en la tabla de la *figura 7.1*.

$a_3$	$a_2$	$a_1$	$a_0$	$f$
0	0	0	0	$0$
0	0	0	1	$\overline{x_1} \overline{x_0}$
0	0	1	0	$\overline{x_1} x_0$
0	0	1	1	$\overline{x_1}$
0	1	0	0	$x_1 \overline{x_0}$
0	1	0	1	$\overline{x_0}$
0	1	1	0	$x_1 \oplus x_0$
0	1	1	1	$\overline{x_1} + \overline{x_0}$
1	0	0	0	$x_0 x_1$
1	0	0	1	$x_1 \oplus x_0$
1	0	1	0	$x_0$
1	0	1	1	$x_0 + \overline{x_1}$
1	1	0	0	$x_1$
1	1	0	1	$x_1 + \overline{x_0}$
1	1	1	0	$x_1 + x_0$
1	1	1	1	$1$

**Figura 7.1.** Tabla que representa las 16 funciones lógicas de dos variables, en función del valor de los coeficientes ( $a_0, \dots, a_3$ ) de su representación por minterms.

Así pues, si queremos tener una función universal de 2 variables,  $(x_1, x_0)$ , empezaríamos invirtiéndolas para obtener  $\overline{x_1}$  y  $\overline{x_0}$ . Después construiríamos los cuatro términos producto  $(x_1 x_0, x_1 \overline{x_0}, \overline{x_1} x_0, \overline{x_1} \overline{x_0})$  y, finalmente, controlaríamos el paso de cada uno de estos términos mínimos a un sumador bajo el control del valor de las coordenadas del vector lógico que especifica cada una de las 16 funciones concretas, tal como se muestra en la *figura 7.2*.

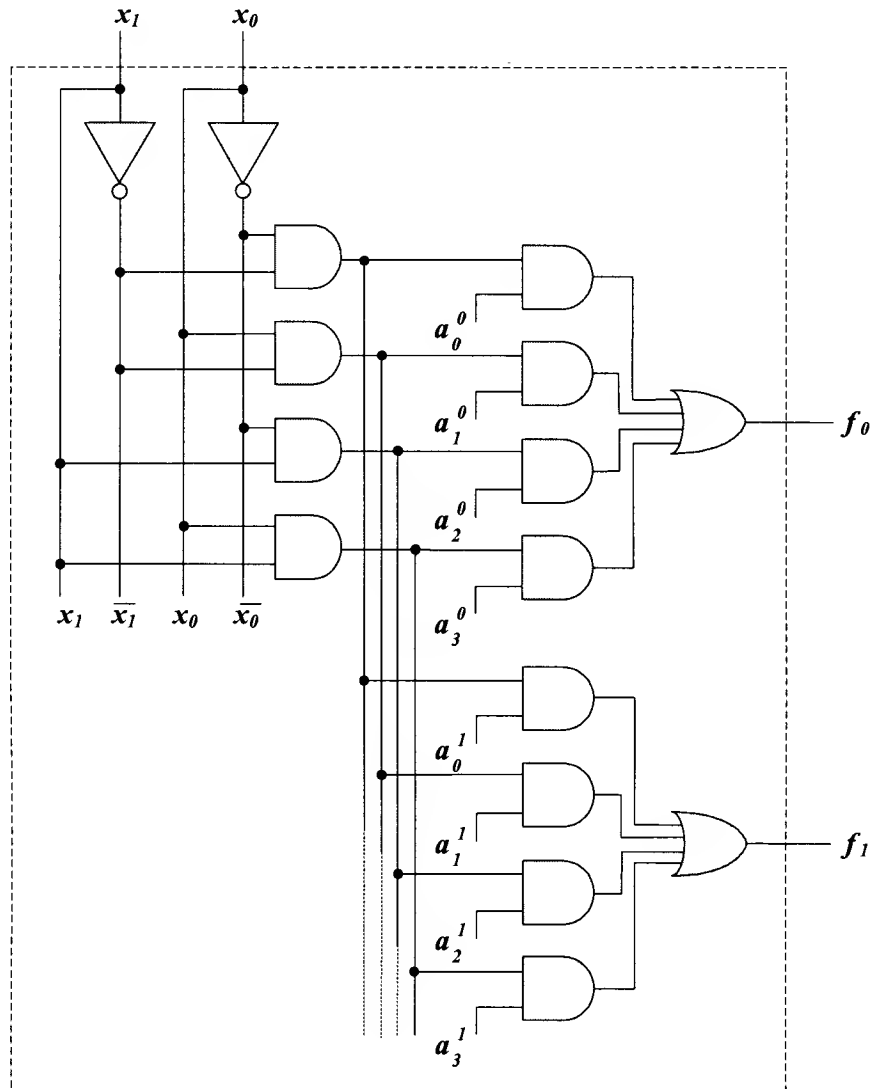
Si quisiéramos obtener *más de una función a la vez* (función múltiple  $f_0, f_1$ , por ejemplo), tendríamos que añadir otra puerta OR a la salida y los correspondientes circuitos de control a partir de los mismos términos mínimos, tal como se muestra en la *figura 7.3*.



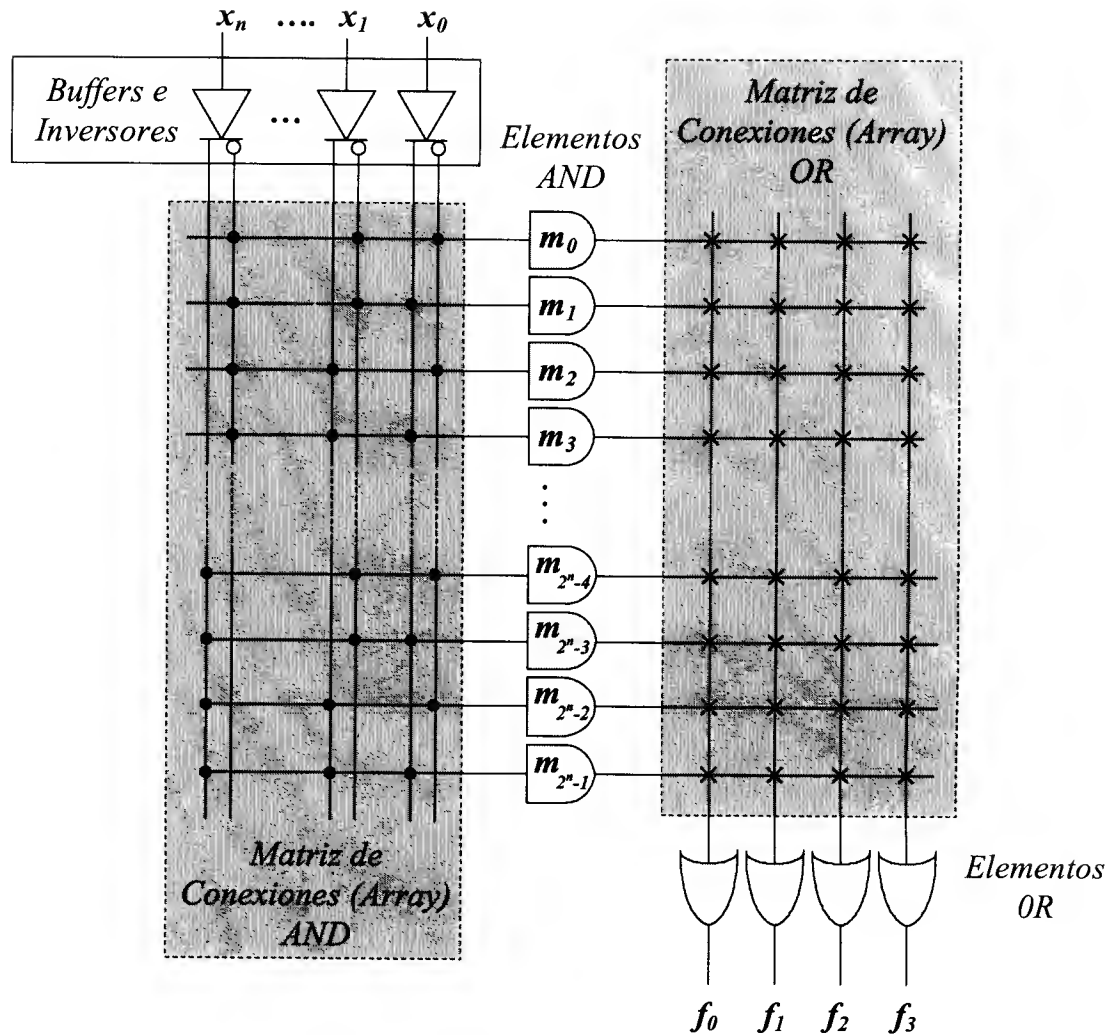
**Figura 7.2.** Función lógica universal de dos variables. Cambiando el valor de los cuatro términos de control ( $a_0, \dots, a_3$ ) recorreremos las 16 funciones posibles.

Para duplicar el número de funciones que comparten un mismo conjunto de términos mínimos hemos duplicado los vectores de control ( $a_0, a_1, a_2, a_3$ ). Ahora cada nueva puerta OR tiene acceso a todos los términos producto (filas) pero necesita una columna de conmutadores programables para controlar cuales son, de hecho, los términos que, en un determinado patrón de configuración, participan en su función. Así pues, la generalización de este procedimiento de síntesis se obtiene con dos matrices de conexiones programables. Una matriz AND que se encarga de generar los  $2^n$  términos mínimos de las  $n$  variables de entrada ( $x_0, \dots, x_n$ ) y otra matriz OR que se encarga de permitirnos agrupar de una forma cómoda cualquier subconjunto de esos  $2^n$  términos mínimos para generar

cualquiera de las  $2^{2^n}$  funciones lógicas diferentes de esas  $n$  variables de entrada. Así aparece la estructura general de toda la lógica combinacional programable que es la que se muestra en la *figura 7.4*, a la que hemos añadido una etapa inicial de amplificadores digitales (buffers).



**Figura 7.3.** Uso de los mismos términos mínimos para obtener a la vez dos funciones universales de 2 variables ( $f_0, f_1$ ).



**Figura 7.4.** Estructura general de la lógica programable (PLDs) con las dos matrices (AND y OR) de conexiones programables y las dos ristas de elementos AND y OR.

Hay tres arquitecturas básicas de PLDs: la PROM ("Programmable Read Only Memory"), la PAL ("Programmable Array Logic") y la PLA ("Programmable Logic Array"). Las tres comparten el esquema general de la *figura 7.4*, con buffers e inversores para adaptar la impedancia de entrada y para generar las negaciones de todas las variables de entrada. Después se generan los términos productos y, finalmente, a través de la matriz OR se posibilita la participación de esos términos producto en cada

una de las  $m$  líneas de salida. Las diferencias básicas entre estas tres arquitecturas PLDs (PROMs PALs y PLAs) están en el carácter fijo o programable de cada una de estas matrices:

<i>Arquitectura</i>	<i>Matriz AND</i>	<i>Matriz OR</i>
<b>PROM</b>	<i>fija</i>	<i>programable</i>
<b>PAL</b>	<i>programable</i>	<i>fija</i>
<b>PLA</b>	<i>programable</i>	<i>programable</i>

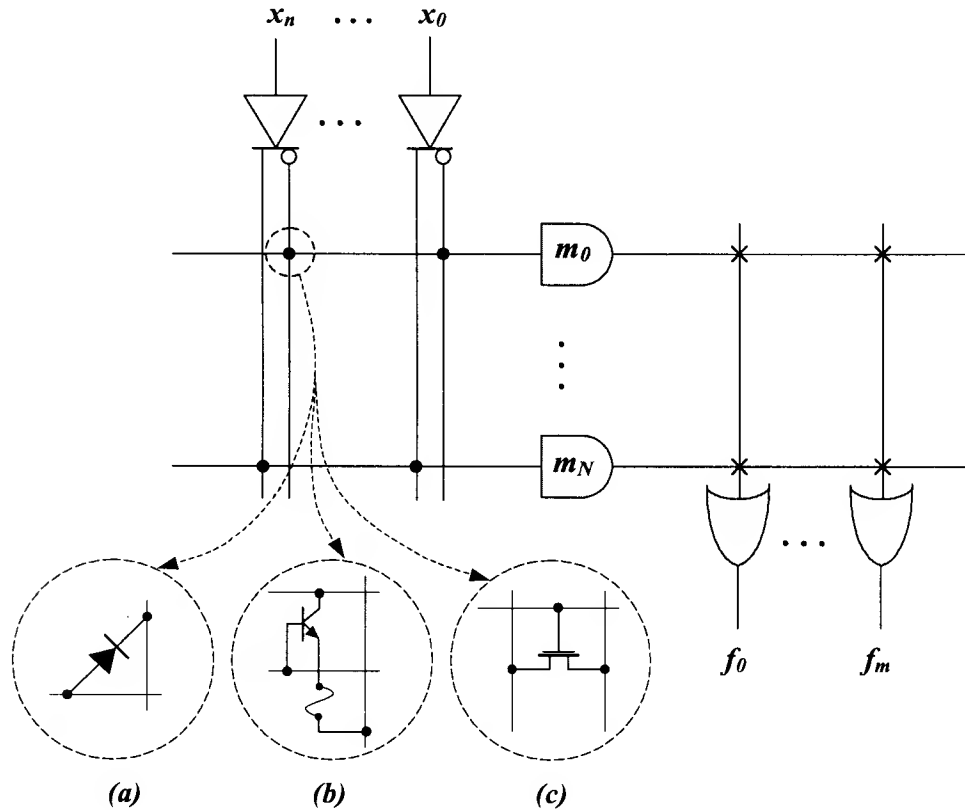
Así, la arquitectura PROM tiene todos los términos mínimos (se usen o no) y sólo es programable la parte OR, por lo que se pueden obtener tantas funciones de salida como líneas OR tenga la PROM. En cambio la arquitectura PAL deja fija la parte OR, limitando el número de líneas AND que pasan a ser sumadas. Finalmente la arquitectura PLA es la más general pues deja programables los términos mínimos y los coeficientes de las sumas. Cada una de estas arquitecturas encuentra un repertorio de aplicaciones para las que es más idónea.

La evolución tecnológica de las distintas arquitecturas en lógica combinacional programable está asociada a los distintos procedimientos usados para programar y reprogramar (en su caso) los distintos nudos (conmutadores de cruce de filas y columnas) que definen la conectividad efectiva en las matrices AND y OR y, por consiguiente, la funcionalidad específica del PLD.

Las primeras PROM se programaban por máscaras partiendo de una matriz de diodos y fundiendo las conexiones no deseadas (*figura 7.5.a*). Los diodos pueden sustituirse por cualquier otro dispositivo unidireccional, tal como un transistor bipolar (*figura 7.5.b*) o un transistor de efecto campo (*figura 7.5.c*). En todos los casos el objetivo es controlar cuándo un enlace está presente y cuándo no lo está. Sin embargo, en este tipo de programación por "fusible", las alteraciones producidas son irreversibles. Es decir, los PLD's de esta primera generación, no pueden volverse a programar.

La siguiente etapa fue conseguir que la programación no destruyera nada, de forma que pudiéramos *borrar* una determinada configuración y *reprogramar* el circuito. Para la síntesis de EPROM's (Erasable-PROM), se utilizan estructuras MOS (tipo FAMOS, por ejemplo), en las que la programación se realiza eléctricamente situando un paquete de carga en el nudo de enlace que es un

transistor MOS, y el borrado puede ser eléctrico (EEPROM) o mediante radiación ultravioleta. En ambos casos se elimina la carga almacenada en una capacidad local pero no se destruye la estructura.

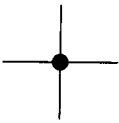
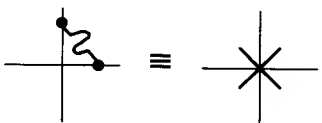
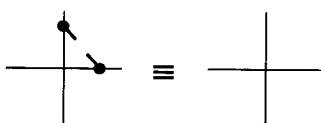


**Figura 7.5.** Distintas formas de sintetizar la programación del nudo de enlace entre filas y columnas en las matrices AND y OR.

Como consecuencia de la explosión combinatoria ( $2^n$  líneas para  $n$  variables) los dibujos de los circuitos PLDs se complican mucho. Por eso se usa un criterio simplificador para representar las conexiones fijas o permanentes, las programables intactas y las programables fundidas. La figura 7.6 resume este criterio de representación. La intersección de dos líneas marcadas con un punto representa una conexión permanente, que no admite programación. Sea cual fuere el procedimiento usado para establecer o eliminar un enlace, cuando la conexión programable está intacta se representa por un aspa y cuando es programable y está fundida se representa por el cruce ortonormal de las líneas

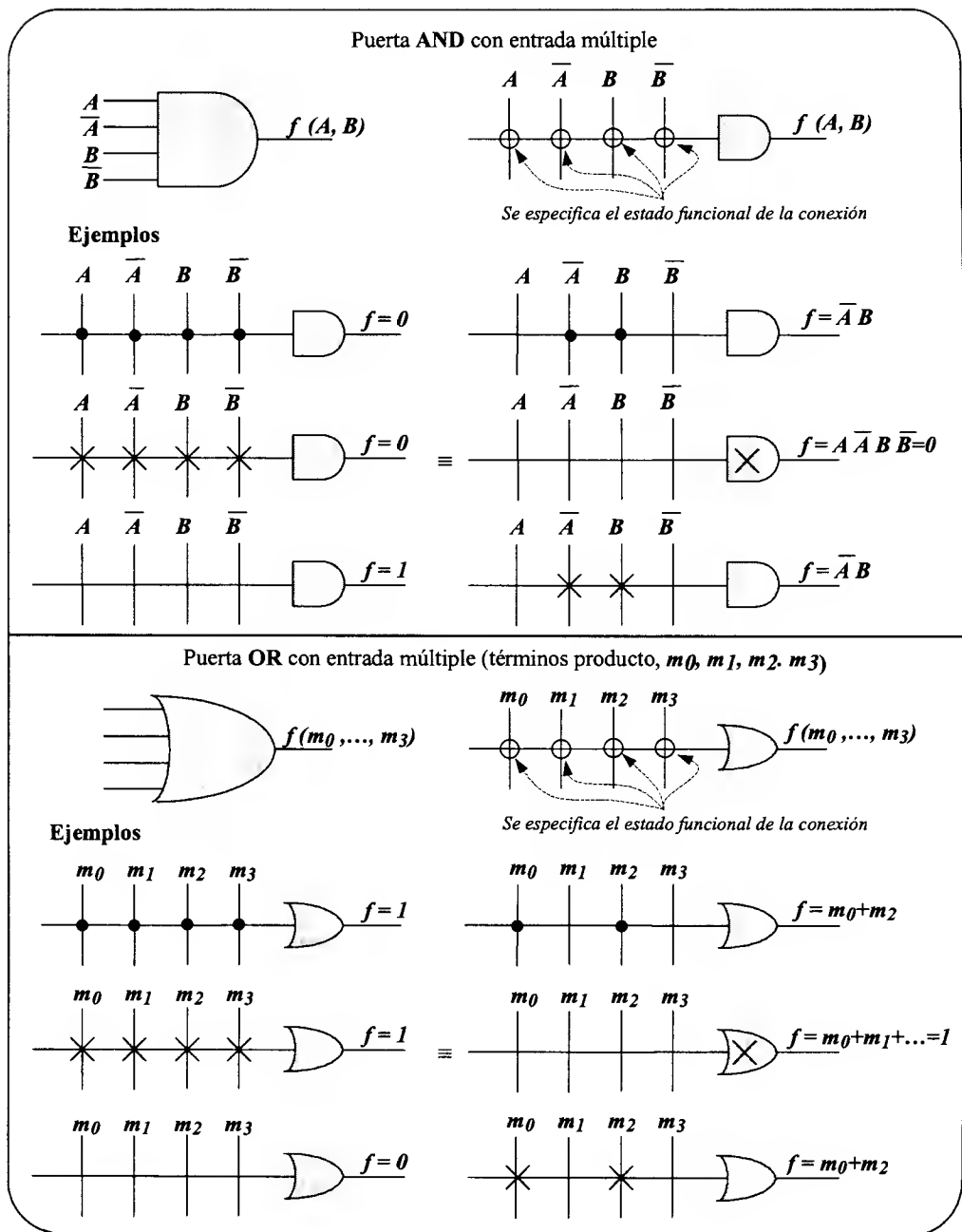


de filas y columnas correspondientes. Obsérvese que decir que una conexión está intacta significa que el enlace ha sido hecho efectivo y, por consiguiente, hay conexión entre las variables que representan la fila y columna correspondientes. En cambio, decir que hemos fundido el enlace significa que, no se ha establecido la conexión tras el proceso de programación. Obviamente, cuando se pasa de programación por fusibles a programación eléctrica y, mas tarde, a programación funcional a partir del estado de una celda de memoria, mantenemos el criterio de representación haciendo siempre referencia al estado funcional de las conexiones, en general, función del tiempo.

<i>Conexión Permanente</i>	<i>Conexión Programable Intacta</i>	<i>Conexión Programable Fundida</i>
		

**Figura 7.6.** Representación del estado funcional de las conexiones en PLDs

La segunda necesidad de simplificación de la representación en PLDs se refiere a las puertas AND y OR con entradas múltiples. Ahora, en vez de usar  $2^n$  líneas de entrada  $(x_0, \bar{x}_0; x_1, \bar{x}_1; \dots)$  a cada una de las puertas AND y  $2^n$  líneas de entrada (términos mínimos  $m_0, m_1, \dots$ ) a cada una de las puertas OR, se dibuja sólo una línea de entrada a cada puerta (AND o OR) y sobre esta línea se sitúan los símbolos correspondientes al estado funcional de las distintas conexiones después del proceso de programación. La *figura 7.7* muestra este criterio de representación y lo ilustra con algunos ejemplos. Cuando el aspa se introduce dentro del símbolo AND significa que todos los términos de entrada están conectados (tienen sus fusibles intactos) y el término producto resultante es cero. Es decir, este término puede ser ignorado en la suma de productos que constituye la salida. Cuando todos los fusibles se han fundido, no hay conexión y el resultado es un uno. Análogamente ocurre para las puertas OR. Un aspa en el interior del símbolo representa que todos los fusibles están intactos y la suma es 1. Inversamente, cuando todas las conexiones a un elemento OR están abiertas (fusibles fundidos) la salida es cero.



**Figura 7.7.** Representación de puertas con entradas múltiples en PLDs.

**Ejercicio:** Representar con el criterio de representación de PLDs resumido en las figuras 7.6 y 7.7 el circuito de la figura 7.2 con  $a_0 = a_3 = 1$  y  $a_1 = a_2 = 0$

**Solución:**

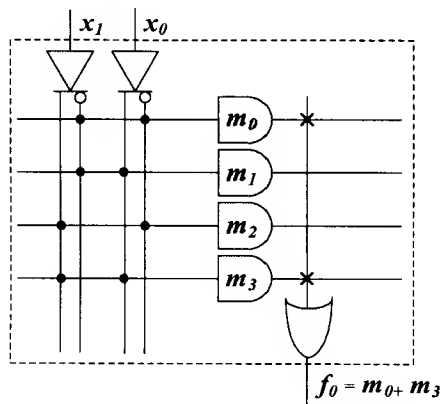


Figura 7.8.

**Ejercicio:** Repetir el ejercicio anterior para el circuito de la figura 7.3 con  $a_0^0 = a_1^0 = 1$ ,  $a_2^0 = a_2^1 = 0$ ,  $a_0^1 = a_3^1 = 1$ ,  $a_1^1 = a_2^1 = 0$ .

**Solución**

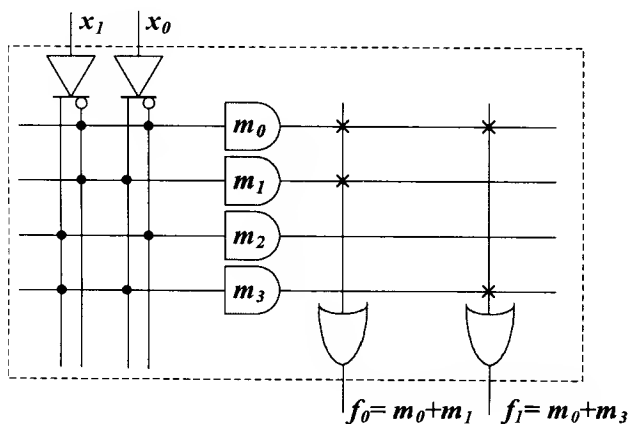


Figura 7.9.

## 7.2. MEMORIAS PROM, EPROM, EEPROM Y FLASH

Bajo el nombre genérico de PROMs englobamos toda la familia de circuitos de memoria no volátil: ROM, PROM, EPROM, EEPROM y FLASH. El origen de todos estos circuitos está en las memorias ROM no volátiles y de sólo lectura que ya venían programadas de fábrica para realizar funciones múltiples tales como convertidores de código, generadores de caracteres, almacenamiento de tablas de datos y almacenamiento de programas. Su limitación fundamental estaba en que una vez programada la memoria ya no podía modificarse el programa.

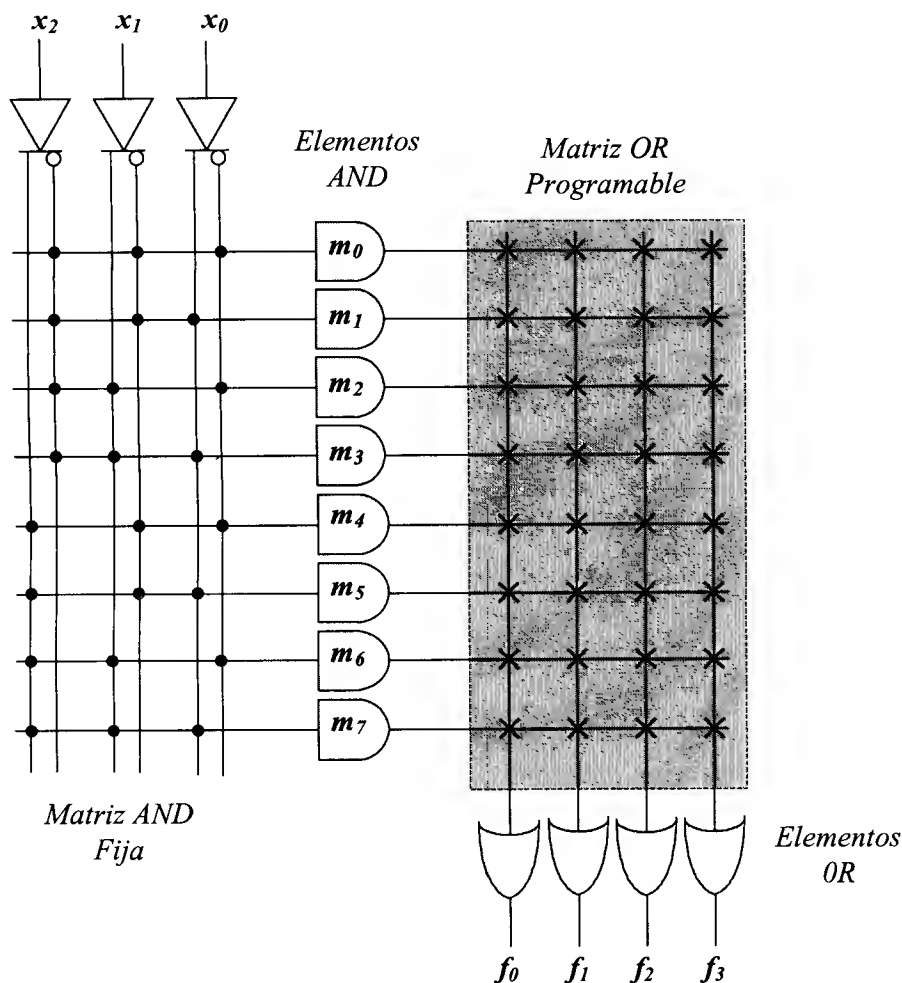
Aparece entonces la necesidad de las PROM que, como hemos visto anteriormente, son PLDs en los que la parte AND es fija y completa, estando así accesibles todos los términos mínimos de las  $n$  variables de entrada, y la parte OR es programable. Ahora es el usuario quien puede programar la PROM, por ejemplo, para almacenar el programa de control de una lavadora o de una máquina herramienta de acuerdo con sus propios algoritmos. Al final de un desarrollo sobre funciones electrónicas de instrumentación o control, reducimos los algoritmos a un programa y este lo grabamos en una PROM, con lo que hemos convertido una arquitectura general en un diseño de propósito especial en el que la parte más genuina del conocimiento está en la EPROM que nosotros hemos grabado.

La *figura 7.10* muestra la arquitectura de una PROM de tres entradas ( $x_0, x_1, x_2$ ) y cuatro salidas ( $f_0, f_1, f_2, f_3$ ). Obsérvese que, al disponer de todos los términos mínimos de tres variables, ( $m_0, m_1, \dots, m_7$ ) para obtener nuevas funciones de salida ( $f_4, f_5, \dots$ ) sólo necesitamos añadir nuevas puertas OR. Es decir, nuevas columnas a la matriz OR. Un inconveniente de las PROM es la necesidad de duplicar la complejidad de la parte fija AND cada vez que queramos añadir una nueva variable de entrada ( $x_3, x_4, \dots$ ). El otro inconveniente de las PROMs es que no son reprogramables y, por consiguiente, no admiten errores.

Así, el siguiente paso en la evolución de las memorias no volátiles PROM fue conseguir que fueran reprogramables. Esto exige que podamos borrar primero el programa que ya tenía almacenado la matriz de conexiones sin destruir la estructura.

Dependiendo del mecanismo usado para borrar tenemos dos nuevos tipos de memorias no volátiles: EPROM y EEPROM (ó E<sup>2</sup>PROM).

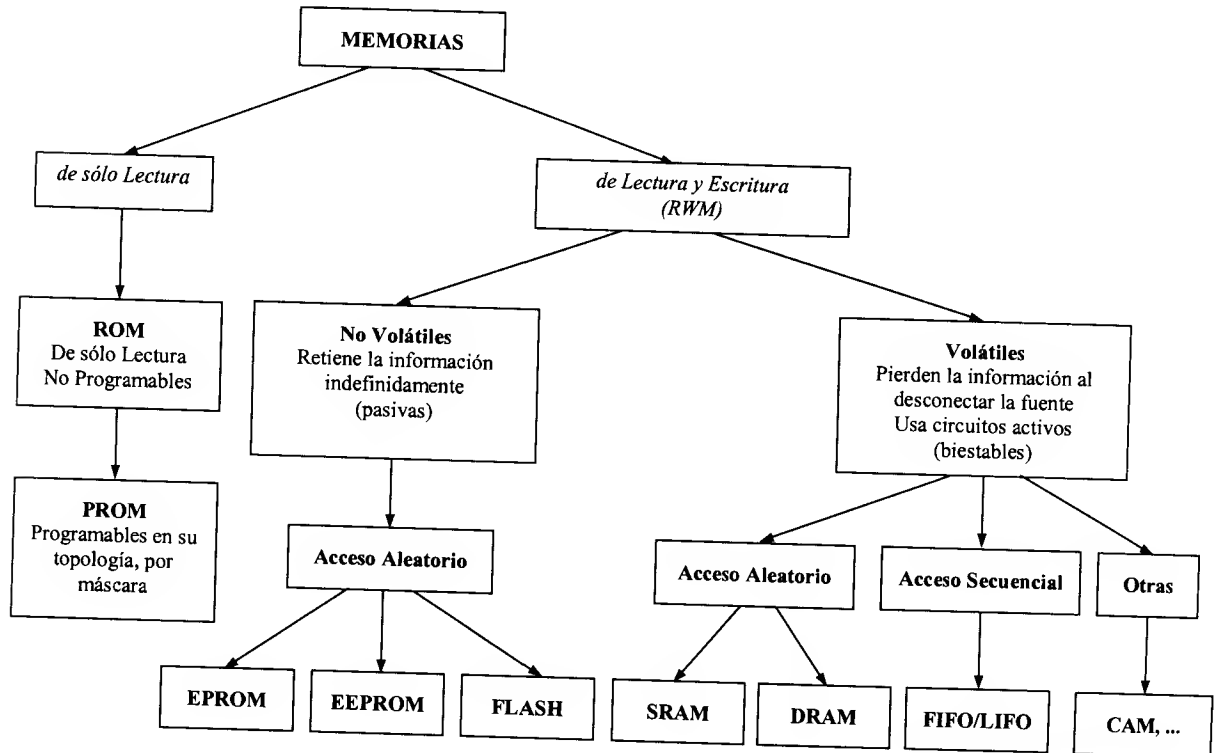
Las EPROM (Erasable PROM) se borran usando luz ultravioleta y se programan eléctricamente y las EEPROM (Electrically Erasable PROM) se borran y se graban eléctricamente, por lo que pueden realizarse ambos procesos sin necesidad de extraer el circuito de memoria del resto de los circuitos que constituían la aplicación.



**Figura 7.10.** Esquema de una memoria PROM de tres variables de entrada ( $x_0$ ,  $x_1$ ,  $x_2$ ) y cuatro funciones programables de salida ( $f_0$ ,  $f_1$ ,  $f_2$ ,  $f_3$ ). Cambiando la naturaleza tecnológica de las cruces de filas y columnas de la parte programable, este mismo esquema es válido para EPROM y EEPROM.

Finalmente, la complejidad de las celdas EEPROM limitaba (en términos relativos) el valor máximo de la densidad de integración y, por consiguiente, la capacidad máxima de los circuitos. En las memorias FLASH se resuelve este problema al mantener las características de almacenamiento no volátil y la posibilidad de borrado y regrabación eléctrica de las EEPROMs pero aumentando la densidad de integración como consecuencia de usar celdas diseñadas usando un único transistor. Esto les permite competir eficazmente (como elemento de almacenamiento permanente) con los discos magnéticos o CDROMs en aplicaciones de computación que solo demandan una capacidad limitada de memoria externa o secundaria.

El esquema de la *figura 7.11*. resume la saga de las memorias ROMs en el contexto general de las memorias de semiconductores que incluyen también a las otras memorias con celdas activas que estudiaremos más tarde, dentro de los circuitos secuenciales (DRAM, SRAM, CAM, FIFO y LIFO).



**Figura 7.11.** Clasificación de los circuitos de memoria.

Obsérvese que en la elaboración de este árbol de clasificación se han usado varios criterios:

1. *Sus funcionalidades.*
2. *La volatilidad de los datos.*
3. *La forma de direccionamiento.*
4. *El mecanismo físico soporte.*

En cuanto a su funcionalidad las memorias se clasifican en “de sólo lectura” (ROM y PROM) y “de lectura y escritura”. A su vez, dentro de estas últimas hay que distinguir entre las no volátiles y las volátiles, haciendo referencia al tiempo durante el cual son capaces de retener la información almacenada cuando se corta la alimentación. Todas las memorias que hemos visto en este apartado (EPROM, EEPROM y FLASH) son de naturaleza no volátil. En cambio, todas las que estudiaremos en lógica secuencial (RAM estáticas y dinámicas, FIFO, etc) son volátiles porque los elementos de almacenamiento (las “celdas”) son circuitos activos, biestables, que necesitan la polarización (la fuente de alimentación) para mantener su estado lógico.

El siguiente criterio de clasificación usado tiene que ver con el direccionamiento, es decir, con el mecanismo mediante el cual se accede al contenido de una celda de memoria. Hay dos formas básicas de acceso, el secuencial y el aleatorio (Random Access Memory) e históricamente se ha asociado el acceso aleatorio con las memorias volátiles construidas con biestables (SRAM). Sin embargo, todas las memorias no volátiles tipo EPROM o FLASH son también de acceso aleatorio. Es decir, cuesta el mismo tiempo acceder a cualquiera de sus direcciones, independientemente del orden y/o contenido de esa dirección.

### **7.3. TRANSISTORES DE PUERTA FLOTANTE (FAMOS) Y MECANISMOS DE BORRADO**

El último criterio de distinción dentro del grupo de las memorias no volátiles con posibilidad de lectura y escritura es el mecanismo físico soporte de los distintos procesos electrónicos asociados al almacenamiento permanente, la programación y el borrado de la memoria.

La arquitectura de todas las memorias no volátiles es esencialmente análoga a la de las ROM y PROM y está formada por una matriz de transistores situados en los puntos de cruce de *las líneas de palabra* con las *líneas de bit* de forma que la programación de la memoria se realiza facilitando o

inhibiendo de forma selectiva cada uno de esos transistores. La diferencia esencial aquí es que este proceso se tiene que realizar eléctricamente y tiene que ser reversible. Para ello utiliza un transistor MOS con una estructura especial que incluye una puerta flotante donde se almacena un paquete de carga representativo del estado funcional del nodo. Así, el cero se representa por un paquete de carga negativa y el uno por un nivel de carga muy inferior. A nivel de comportamiento externo esta carga almacenada en la puerta flotante se manifiesta en un incremento en la tensión umbral del transistor de forma que ahora, cuando el umbral está alto el transistor no conduce y cuando el umbral baja a su valor normal el transistor de puerta flotante se comporta como cualquier otro MOS, conduciendo entre fuente y drenador de acuerdo con el valor de la tensión aplicada en la puerta externa.

La *figura 7.12* muestra el esquema de un corte transversal de la estructura física de un transistor de puerta flotante. Este dispositivo fue propuesto por Kahng y Sze en 1967, y consiste en una modificación en el electrodo de puerta incorporándole otra capa fina de aislante entre la puerta externa y el canal, de forma que queda otra puerta flotando muy cerca del semiconductor para permitir el transporte de carga cuando se aplica una tensión externa relativamente alta al electrodo de puerta, generando un proceso de inyección de carga por avalancha desde el sustrato. Esta carga almacenada es la responsable del incremento del umbral y permanece durante muchos años. Para eliminarla (para borrar la memoria) se usa distintos procedimientos, que son los que distinguen las EPROM (ultravioleta) de las EEPROM y las FLASH (tensiones eléctricas de signo opuesto al de las de propagación).

La idea de puerta flotante se desarrolló en dos grupos de dispositivos:

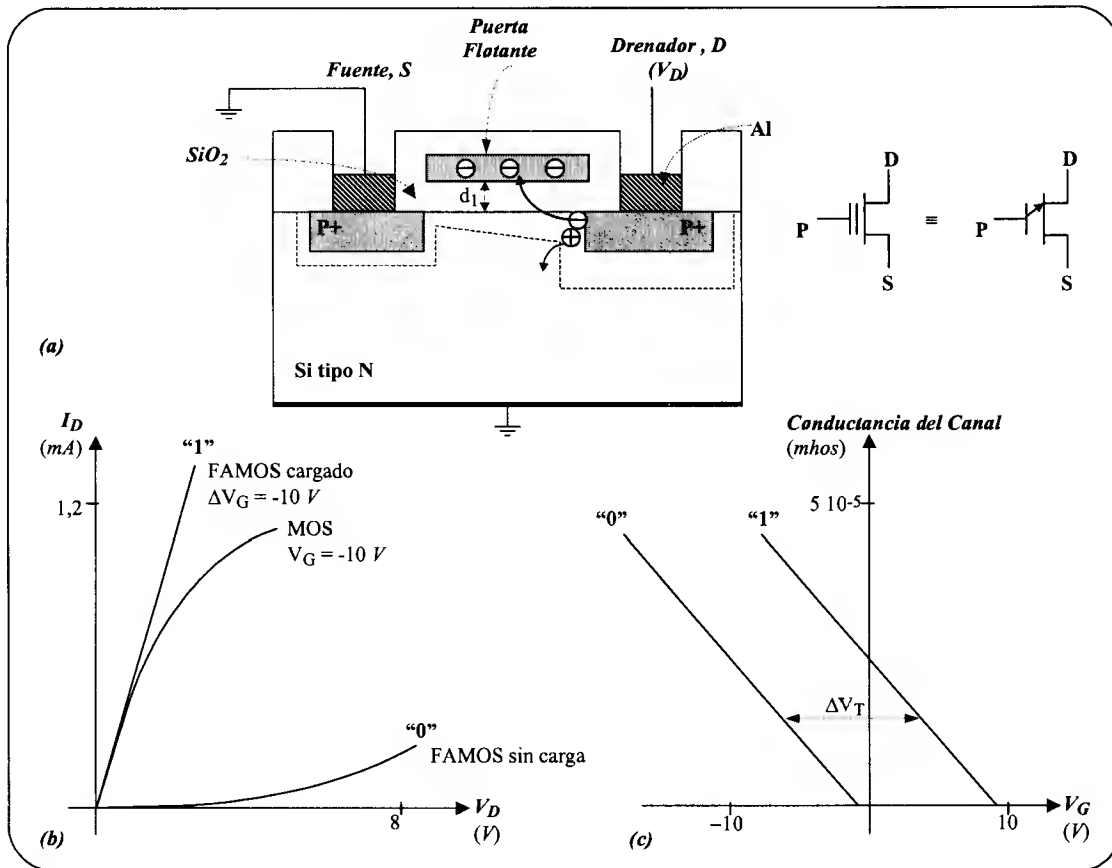
1. Los FAMOS (Floating-gate Avalanche injection MOS memory)
2. Las estructuras MIOS (Metal Insulator Oxide Semiconductors).

De ellas, ha sido la FAMOS (Frohman y Bentchkomsky, 1970) la que ha mostrado un mayor desarrollo al combinar la puerta flotante de silicio policristalino con el mecanismo de inyección por avalancha de electrones desde la unión  $P^+N$  para potenciales del orden de 20 V. La carga transferida es función de la amplitud y duración del pulso de programación. Una vez que desaparece este potencial no existe ninguna vía de descarga para los electrones pues la puerta está rodeada de óxido de silicio ( $SiO_2$ ) que es un excelente aislante.

Es decir, la avalancha en  $P^+N$  genera pares electrón-hueco en la zona de transición. Los electrones pasan por efecto túnel a la puerta flotante y allí quedan. Los huecos caminan hacia el



sustrato y contribuyen a la creación de una capa de inversión entre drenador y fuente en las proximidades de la superficie. A este efecto contribuye también un mecanismo de inducción. La carga negativa en la puerta flotante induce carga positiva en la superficie.



**Figura 7.12.** Estructura FAMOS y SAMOS. (a) Concepto de puerta flotante en FAMOS, de acuerdo con la propuesta inicial de Kahng y Sze. Modificación de Lizuka. Valores de  $g_D$  (b) y  $\Delta V_T$  (c).

La cantidad de carga transferida a la puerta es función de la amplitud y duración del potencial aplicado a la unión P<sup>+</sup>N y se puede medir a través de la medida del cambio en la conductividad del canal superficial. Recordemos que la conductancia del canal,  $g_D$ , venía dada por la expresión del tipo:

$$g_D = -\frac{Z}{L} \mu_T C_{ox} (V_G - V_T), \quad V_G > V_T \quad \text{canal P} \quad [7.2]$$

La carga negativa de la puerta ha producido un cambio en  $V_T$  de valor:

$$\Delta V_T = -\frac{Q_G}{C_{ox}}, \quad [7.3]$$

por lo que se ha producido también un cambio en  $g_D$ . La parte (b) de la *figura 7.12* muestra las curvas ( $I_D$ ,  $V_D$ ) y las variaciones de  $g_D$  para un MOS normal y para una estructura FAMOS con carga y sin carga en la puerta aislada. La parte (c) muestra a variación de  $g_D$  con  $V_G$  y el valor de  $\Delta V_T$  entre los estados “0” y “1”.

### 7.3.1. Borrado de EPROMs

Para que la estructura FAMOS pueda cumplir funciones de memoria es necesario que podamos borrarlas (eliminar la carga almacenada en las puertas flotantes). Como no hay puertas externas, este borrado no puede ser por procedimientos eléctricos. Así, las EPROM se borran iluminando el circuito con luz ultravioleta. Esta radiación hace que el óxido sea ligeramente conductor debido a la generación de pares electrón-hueco en el material. Este proceso puede llevar varios minutos frente a los pocos microsegundos que son necesarios para la grabación.

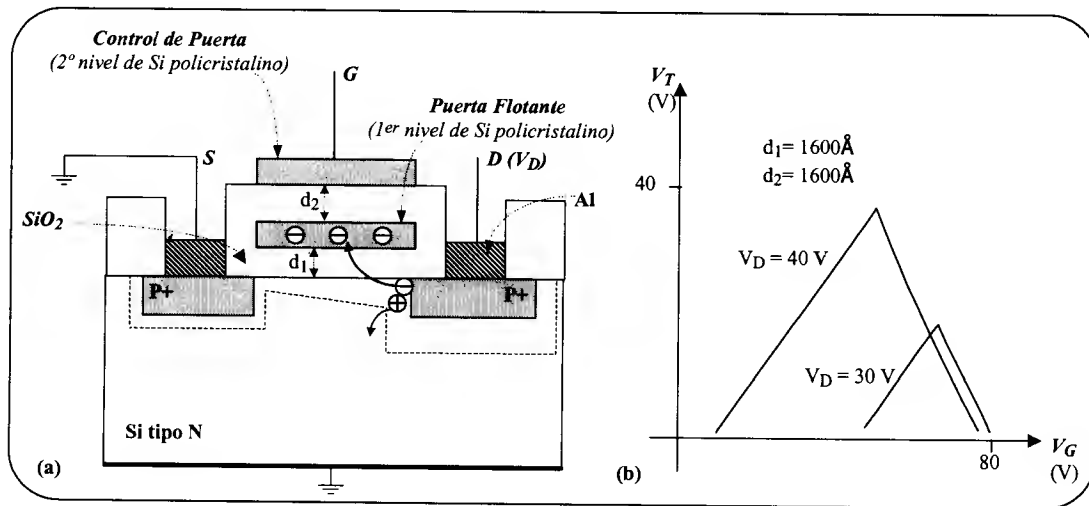
Otro problema es la degradación de la estructura que limita el número máximo de veces que puede borrarse y reprogramarse la memoria. Finalmente, también un gran inconveniente es el hecho de que para borrar la EPROM tengamos que quitarla del sistema en el que está integrada y llevarla a un programador externo para programarla.

Las ventajas sin embargo están en la sencillez de las celdas EPROM lo que permite su fabricación con alta densidad de integración y bajo coste, lo que las hace adecuadas para aquellas aplicaciones que necesitan gran cantidad de memoria pero que no requieren una reprogramación excesivamente frecuente.

### 7.3.2. Borrado de las EEPROMs

Para conseguir borrar eléctricamente las PROMs tenemos que incluir un electrodo externo y un mecanismo que permita eliminar la carga de las puertas aisladas sin necesidad de desconectar la memoria del resto del sistema.

Históricamente esto se consiguió con las estructuras SAMOS (Staked-gate Avalanche-injection MOS) propuestas por Lizuka en 1976 e ilustradas en la *figura 7.10*. Esta estructura posee un doble nivel de puertas de silicio policristalino de forma que la puerta externa (llamada de control) permite el borrado eléctrico. Tras la inyección por avalancha, las cargas negativas se almacenan en la puerta flotante causando el cambio en el valor de la tensión umbral,  $V_T$ , del MOS (*estado "1"*). Después, aplicando una tensión positiva grande a la puerta externa, la carga almacenada se elimina y el MOS vuelve al estado "0".

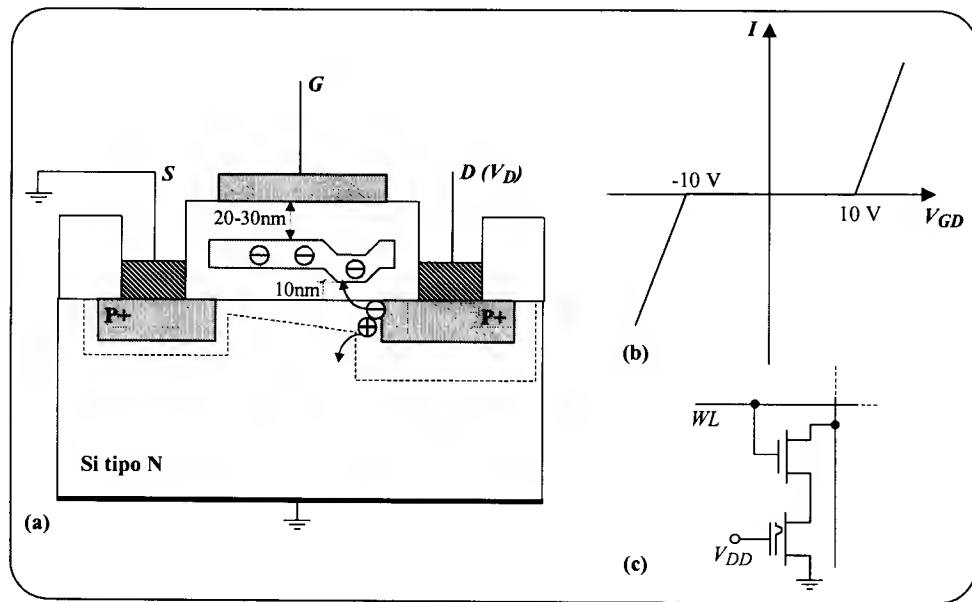


**Figura 7.13.** (a) Estructura SAMOS con doble puerta para permitir el borrado eléctrico. (b) Variación del umbral (forma cualitativa) en función del potencial en la puerta de control para distintos valores de la tensión de drenador.

La parte *b* de la *figura 7.13* muestra las características de escritura en SAMOS. Para una determinada tensión de drenador,  $V_T$  aumenta con  $V_G$  porque esta tensión en la puerta externa ayuda al transporte de los electrones, generados por avalancha desde el silicio hasta la puerta flotante, igual que en FAMOS. Sin embargo, a partir de un cierto valor de  $V_G$ , al seguir aumentando hace que  $V_T$  disminuya porque el transporte de corriente desde la puerta flotante a la puerta externa hace que disminuya la carga almacenada.

Hay otras estructuras, como la FLOTOX (FLOating-gate Tunneling Oxide) que también permiten el borrado eléctrico de las EEPROMs (Jonson, 1980). Tal como se ilustra en el esquema del corte

transversal de la *figura 7.14*, la FLOTOX es muy parecida a la FAMOS sólo que se ha modificado la parte del dieléctrico que separa la puerta flotante del canal en las proximidades del terminal de drenador, haciéndolo más estrecho (menos de  $10\text{ nm}$ ). Así, cuando se aplica una tensión del orden de  $10\text{ voltios}$ , el gran campo eléctrico que se produce sobre esta parte delgada del aislante hace que los electrones puedan pasar a la puerta flotante en ambos sentidos por un mecanismo de efecto túnel, dando lugar a las características tensión-corriente de la unión túnel que se muestra en la parte (b) de la *figura 7.14*. Y este proceso es reversible. Para borrar la memoria basta con invertir el signo del potencial usado para escribir.



**Figura 7.14.** (a) Estructura FLOTOX programable por efecto túnel. (b) Características tensión corriente, donde se observa que no conduce entre  $-10$  y  $+10$  voltios y que se comporta después como un diodo en directa. (c) Celda EEPROM tal como se configura durante la operación de lectura.

Como en el resto de los transistores de puerta flotante, la inyección de electrones aumenta el umbral  $V_T$  mientras que la extracción lo disminuye. Sin embargo, el valor final del umbral, tras extraer o inyectar electrones, dependerá del estado inicial de carga de la puerta flotante por lo que aparece un problema de control del umbral que se resuelve añadiendo un nuevo transistor que actúa como dispositivo de acceso durante la operación de lectura, mientras que el transistor FLOTOX actúa

como elemento de almacenamiento. Esto hace que las EEPROMs tengan celdas más complicadas que las EPROMs porque el transistor FAMOS hacía a la vez ambas funciones, la programación y el almacenamiento. La ventaja de las EEPROMs está en su versatilidad y en la posibilidad de ser borradas y reprogramadas muchas más veces que las EPROM y sin sacar el circuito de la placa.

### 7.3.3. Borrado de las memorias FLASH

Las memorias FLASH EPROM combinan la densidad de integración propia de los transistores FAMOS (EPROM) y la versatilidad de las EEPROMs porque integran los dos mecanismos básicos de inyección y extracción de carga en la puerta flotante. Así, en general, usan el mecanismo de inyección por avalancha para la programación de los distintos MOS y el mecanismo de efecto túnel para borrar la memoria. Así, la estructura física de una memoria flash es muy parecida a la de los transistores FAMOS sólo que ahora la capa de óxido que separa la puerta flotante del canal es muy delgada ( $10nm$ ) para facilitar el efecto túnel. Además, se usan áreas diferentes de la puerta para la programación y el borrado. Para programar la celda se aplican  $12\text{ voltios}$  a la  $V_G$  y  $V_D$ , con  $V_S = 0$ . Para borrar, se pone  $V_G = 0$  y  $V_S = 12\text{ voltios}$  (figura 7.15).

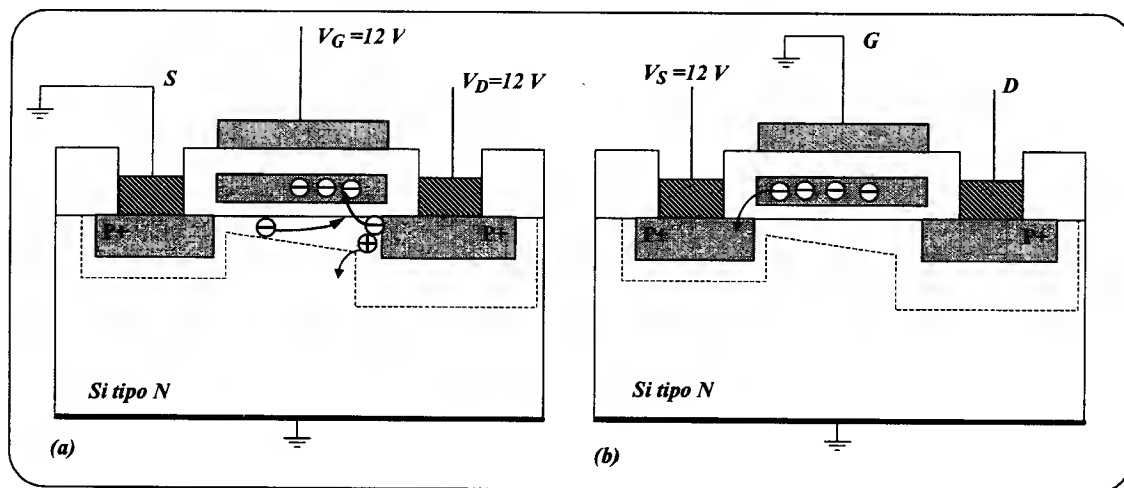


Figura 7.15. Memoria FLASH. (a) Programación. (b) Borrado

Una característica distintiva de las memorias FLASH es que el borrado se realiza a la vez sobre toda la memoria o por bloques. Además, estas incluyen circuitos adicionales a los de grabación y

borrado que tienen como finalidad monitorizar el valor de los umbrales durante el borrado para ajustar dinámicamente el tiempo de borrado y reducir al mínimo la necesidad de señales externas para su acoplo a microprocesadores. Este tipo de diseño que incluye en el “chip” los circuitos de monitorización de umbrales en el proceso de borrado sólo es práctico cuando se borran de una vez subsecciones importantes de la memoria. La tabla de la *figura 7.16* compara algunas características de las memorias no volátiles reprogramables. De su consideración parece evidente la ventaja de las EEPROM tipo flash por su alta capacidad de almacenamiento (como las EPROMs), la posibilidad de lectura-escritura en la placa en la que está integrada (como las EEPROMs) y por su velocidad y coste. En particular, para aquellas aplicaciones que, junto a las funcionalidades anteriores, exijan una alta flexibilidad. Es decir, un valor alto para el número máximo de ciclos de lectura-escritura sin degradación de los umbrales.

	EPROM	EEPROM	FLASH
♦ <i>Densidad</i>	Alta (1 transistor/celda)	Media (más de 1t/c)	Alta (1t/c)
♦ <i>Tamaño</i>	$\geq 16\text{Mbit}$	$\geq 1\text{Mbit}$	$\geq 16\text{Mbit}$
♦ <i>Tiempos de Acceso</i>	Bajo (60ns)	Medio (120 ns)	Bajo (50 ns)
♦ <i>Tiempo de Borrado</i>	Medio (minutos)	—	Muy bajo (<4 s)
♦ <i>Tiempo de Programación</i>	Muy bajo (<5 $\mu\text{s}$ )	Alto ( $\approx 4\text{ s}$ /circuito)	Muy bajo (<5 $\mu\text{s}$ )
♦ <i>Reprogramable dentro del Sistema</i>	No	Si	Si
♦ <i>Flexibilidad</i> (Nº de ciclos de lectura/escritura sin degradación)	100	10 <sup>5</sup>	10 <sup>3</sup> -10 <sup>5</sup>

*Figura.7.16.* Tabla de características de los distintos tipo de memorias no volátiles con posibilidad de lectura/escritura (adaptada a partir de la de J.M. Rabaey, 1996).

## 7.4. ORGANIZACIÓN INTERNA Y EJEMPLOS DE EEPROM Y FLASH.

Hemos visto el concepto de PROM en el contexto de la lógica combinacional programable y los distintos tipos de memorias no volátiles con capacidad de lectura y reprogramación junto con los mecanismos físicos que hacen posible esa reprogramación eléctrica (transistores FAMOS). Ahora vamos a ver la organización interna adicional a la matriz de transistores y la descripción funcional,

externa, de algunos circuitos comerciales concretos. No son importantes los ejemplos en sí, sino su capacidad para ilustrar los contenidos usuales en las hojas de datos de los distintos fabricantes.

Empecemos con la estructura interna. Todos los circuitos EPROM, EEPROM y FLASH incluyen, junto a la matriz de celdas, otras facilidades adicionales para controlar el proceso de lectura del estado lógico de esas celdas y su reprogramación fuera o dentro del sistema digital en el que operan. Así, son necesarios:

- Circuitos para direccionar las celdas en las que se desea actuar.
- Circuitos para controlar la selección de “chip” y la ejecución del proceso que se desea realizar en la celda seleccionada en ese “chip”.
- Un conjunto de amplificadores digitales (“buffers”) y registros para adaptar las señales de entrada y salida en sus conexiones al bus.
- Otros circuitos especiales en el caso de las EEPROM FLASH.

### 7.4.1. EPROMs

En la *figura 7.17* se muestra un resumen de datos de catálogo de la PROM borrable mediante radiación ultravioleta TMS27C64. Es una EPROM de carácter comercial, perteneciente a la familia 27, es borrable, la tecnología es CMOS y la densidad es de 64K organizada en 8K palabras de 8 bits. Se alimenta a 5 *voltios* y todos sus terminales son compatibles con los niveles de tensión e impedancias de la familia TTL. Su tiempo de acceso es del orden de 100 $ns$ , está preparada para una programación rápida mediante impulsos, posee buffers de tres estados para facilitar la conexión de sus salidas a sistemas organizados por “buses” y su inmunidad al ruido en continua para cargas TTL es de 400 $mvoltios$ .

Esta EPROM admite siete modos de operación de acuerdo con la tabla incluida en la *figura 7.18*. Cuando las salidas de dos o más de estos circuitos están conectadas en paralelo sobre el mismo bus, se puede seleccionar la PROM que queremos leer sin interferir con el resto, poniendo en baja los terminales  $\overline{E}$  y  $\overline{G}$  (circuito seleccionado). A la vez, el resto de los circuitos deben estar inhibidos, con señales en alta aplicadas a uno de estos terminales.

En la tabla de modos de programación de la memoria no incluimos un modo adicional llamado de “firma”, que proporciona el acceso a la información que identifica al fabricante y al tipo de circuito. Este modo se activa cuando  $A_9$  se conecta a 12 *voltios*.

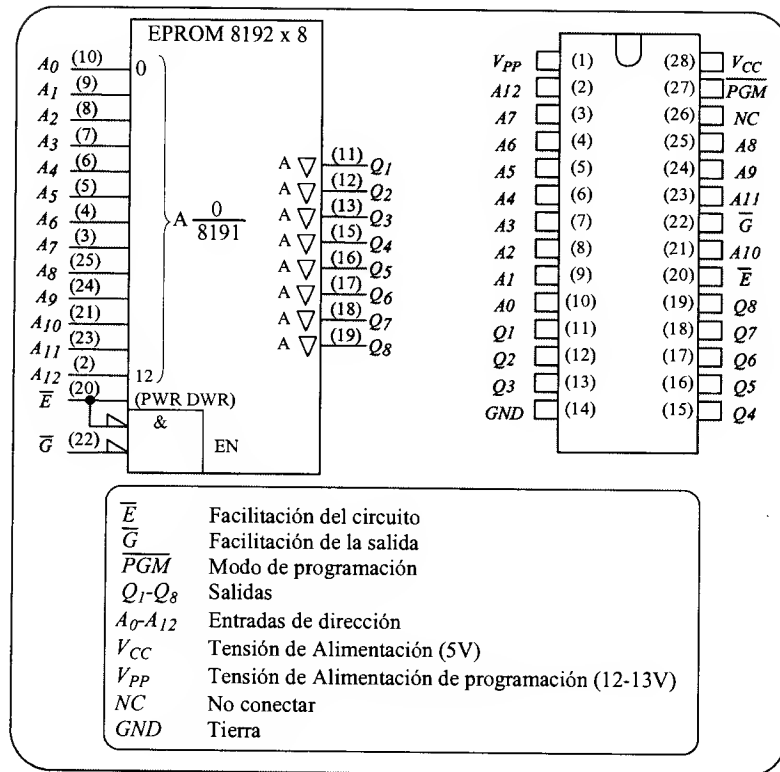


Figura 7.17. Símbolo lógico y significado de los terminales de la memoria EPROM, TMS27C64.

FUNCIÓN	MODOS					
	Lectura	Salida Inhibida	Standby	Programación	Verificación	Inhibición del Programa
$\overline{E}$	V <sub>IL</sub>	V <sub>IL</sub>	V <sub>IH</sub>	V <sub>IL</sub>	V <sub>IL</sub>	V <sub>IH</sub>
$\overline{G}$	V <sub>IL</sub>	V <sub>IH</sub>	x	V <sub>IH</sub>	V <sub>IL</sub>	x
PGM	V <sub>IH</sub>	V <sub>IH</sub>	x	V <sub>IL</sub>	V <sub>IH</sub>	x
V <sub>PP</sub>	V <sub>CC</sub>	V <sub>CC</sub>	V <sub>CC</sub>	V <sub>PP</sub>	V <sub>PP</sub>	V <sub>PP</sub>
V <sub>CC</sub>	V <sub>CC</sub>	V <sub>CC</sub>	V <sub>CC</sub>	V <sub>CC</sub>	V <sub>CC</sub>	V <sub>CC</sub>
A <sub>9</sub>	x	x	x	x	x	x
A <sub>0</sub>	x	x	x	x	x	x
Q <sub>1</sub> - Q <sub>8</sub>	D <sub>out</sub>	alta Z	alta Z	D <sub>in</sub>	D <sub>out</sub>	alta Z

Figura 7.18. Tabla de modos de programación de la Memoria EPROM TMS27C64.

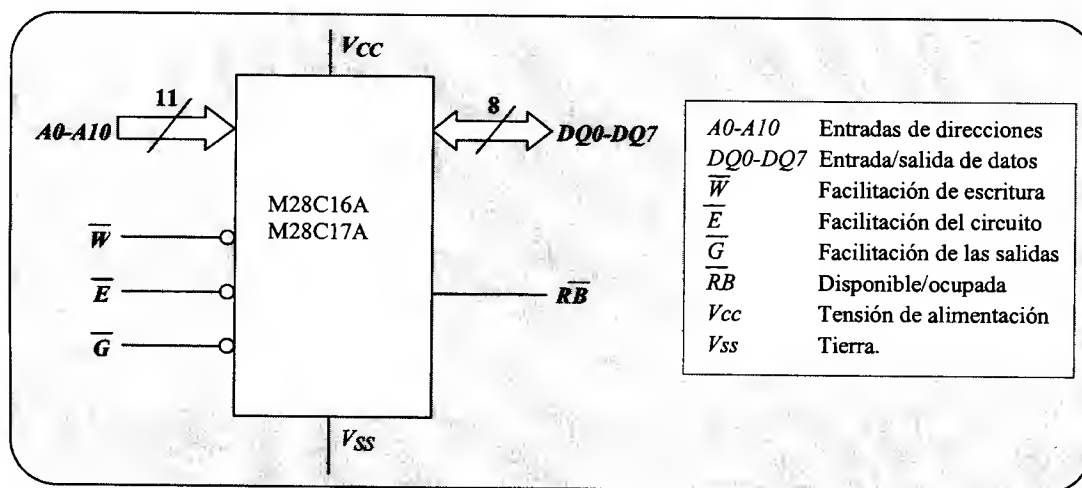


Antes de programar la EPROM, debe borrarse por exposición a luz ultravioleta. Así nos aseguramos que todos los bits están en alta ("1"). Los estados de baja ("0") se programan entonces en las posiciones deseadas, recorriendo todas las direcciones y todas las posiciones dentro de cada dirección. El tiempo de programación es del orden de 1 *segundo*.

Los datos se presentan en paralelo en los terminales ( $Q_1, \dots, Q_8$ ). Una vez que la dirección en la que se va a grabar y los datos están estables, se pulsa  $\overline{PGM}$ . El programador, bajo este algoritmo, genera pulsos de 100  $\mu s$  y verifica que la palabra seleccionada ha sido grabada correctamente, es decir que el contenido de memoria coincide con los 8 bits que están en los terminales  $Q_1, \dots, Q_8$ . Se está en modo de programación, de acuerdo con la tabla de la *figura 7.18*, cuando  $V_{PP} = 13$  voltios,  $V_{CC} = 6.5$  voltios,  $\overline{G} = V_{IH}$  y  $\overline{E} = V_{IL}$ . Cuando se han programado todas las direcciones, se comprueba el estado de todos los bits con  $V_{CC} = V_{PP} = 5$  voltios.

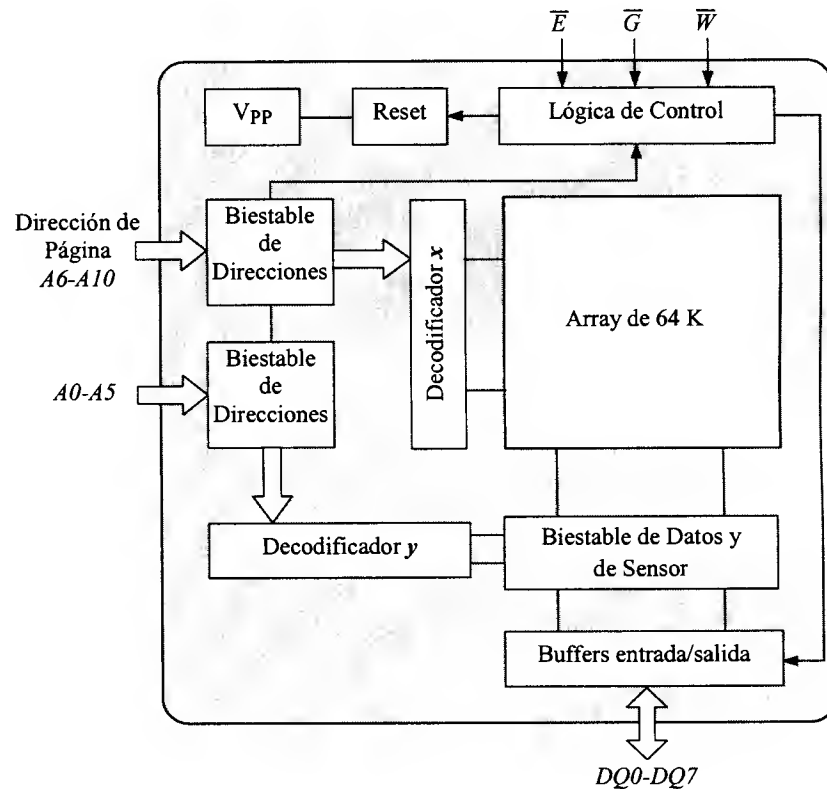
### 7.4.2. EEPROM

La *figura 7.19* muestra el diagrama lógico de la EEPROM de 16 Kbits (2Kx8), M28C16A, organizada en 2K palabras de 8 bits, sobre la que vamos a ilustrar estos circuitos. Con ligeras modificaciones todas las EEPROMs poseen diagramas de bloques, terminales externos y modos de programación análogos.



**Figura 7.19.** (a) Diagrama lógico de la EEPROM M28C16A de ST Microelectronics. (b) descripción del significado de sus terminales

Esta memoria posee un tiempo de acceso de 150ns a 5 voltios y 250ns a 3 voltios. Para operar sólo necesita una fuente de uno de estos dos valores con tolerancias del 10%. El ciclo de escritura es de 5ns y se pueden realizar hasta 100.000 ciclos de borrado/escritura sin degradación, garantizándose la permanencia de la información por más de 40 años. La figura 7.20 muestra el diagrama de bloques de la estructura interna.



**Figura 7.20.** Diagrama de bloques de la estructura interna de la EEPROM M28C16A

Para direccionar las celdas de la EEPROM M28C16A hacen falta 11 bits (A0-A10) de los cuales se gastan 5 para moverse dentro de la página y el resto para la selección de cada una de las páginas. Los terminales DQ0-DQ7 se usan para la entrada y salida de datos tras pasar por los amplificadores. El terminal  $\overline{E}$  (chip enable) debe estar en baja para facilitar todas las operaciones de lectura/escritura y el  $\overline{G}$  (output enable) controla los amplificadores de salida de datos y se usa para iniciar las operaciones de lectura. El terminal  $\overline{W}$  (write enable) controla la escritura de datos y el  $\overline{RB}$

(ready/Busy) es una salida en drenador abierto que puede usarse para detectar el final de un ciclo interno de escritura.

Cuando las señales de control  $\overline{E}$  y  $\overline{G}$  están en baja y  $\overline{W}$  está en alta la EPROM opera en modo de lectura y los datos direccionados se presentan en los terminales de entrada/salida,  $DQ0-DQ7$ . En cambio, cuando bien  $\overline{G}$  o  $\overline{E}$  están en alta, los terminales  $DQ0-DQ7$  permanecen en estado de alta impedancia.

Las operaciones de escritura se inician cuando  $\overline{W}$  y  $\overline{E}$  están en baja los dos y a la vez  $\overline{G}$  está en alta. La dirección queda retenida en el registro en el flanco de bajada de  $\overline{E}$  ó  $\overline{W}$  (el último que se presente) y los datos quedan registrados en el flanco de subida de  $\overline{E}$  ó de  $\overline{W}$  (el que primero se presente).

La organización de la escritura por páginas permite que puedan registrarse en la memoria hasta 32 palabras consecutivas antes de iniciar un ciclo de programación. Todas estas palabras deben estar localizadas en la misma página. Es decir,  $A5-A10$  debe ser el mismo para todas ellas. La tabla de la figura 7.21 resume los valores de  $\overline{E}$ ,  $\overline{G}$ ,  $\overline{W}$  y  $DQ0-DQ7$  correspondientes a los distintos modos de operación de esta EEPROM(\*).

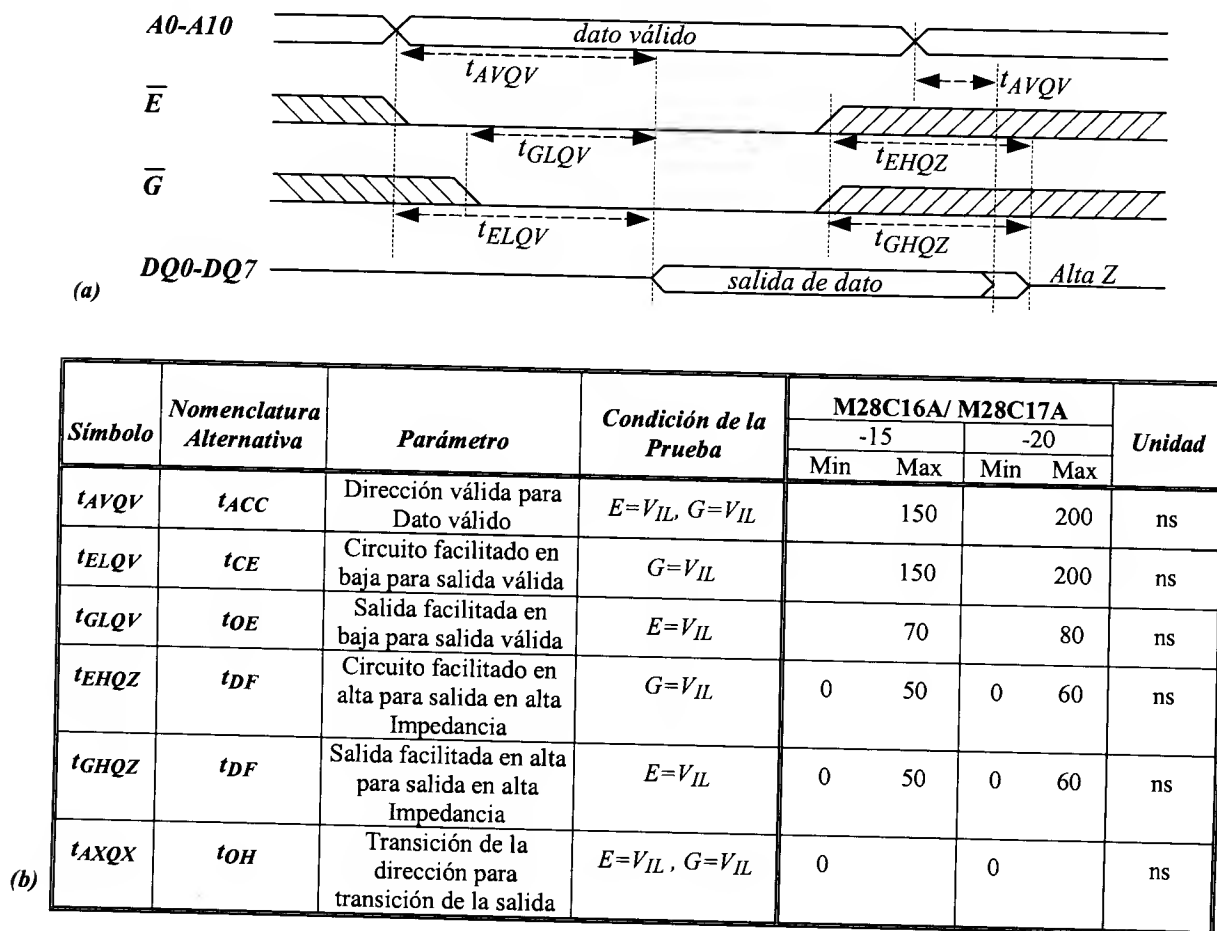
MODO	$\overline{E}$	$\overline{G}$	$\overline{W}$	$DQ0-DQ7$
Lectura	V <sub>IL</sub>	V <sub>IL</sub>	V <sub>IH</sub>	Salida de Datos
Escritura	V <sub>IL</sub>	V <sub>IH</sub>	V <sub>IL</sub>	Entrada de Datos
Standby/Inhibición de Escritura	V <sub>IH</sub>	x	x	Alta Z
Inhibición de Escritura	x	x	V <sub>IH</sub>	Salida de Datos ó Alta Z
Inhibición de Escritura	x	V <sub>IL</sub>	x	Salida de Datos ó Alta Z
Inhibición de Lectura	x	V <sub>IH</sub>	x	Alta Z

**Figura 7.21.** Tabla de valores de las variables de control de la EEPROM M28C16A durante los distintos modos de operación (lectura, escritura).

---

(\*) Aconsejamos visitar la página web de ST (<http://www.st.com>) .

Finalmente, en la *figura 7.22* mostramos las formas de onda correspondientes a las líneas de dirección, datos y señales de control en el modo de lectura, junto con la descripción de los valores correspondientes a los distintos intervalos temporales.



**Figura 7.22.** (a) Forma de onda de los valores estacionarios y las transiciones de las distintas señales de control de la EEPROM durante una operación de lectura. (b) Valores correspondientes de los distintos intervalos temporales

Hemos mantenido en inglés la descripción de los parámetros porque los subíndices de los símbolos correspondientes son los acrónimos de la descripción en lenguaje natural del significado de los parámetros. Así,  $t_{AVQV}$  es el intervalo de tiempo entre el momento en el que tenemos estables los

bits de la dirección (dirección válida) y el momento en el que podemos considerar que también son estables y válidos los datos de salida. Análogamente  $t_{ELQV}$  es el intervalo de tiempo entre el momento en que tenemos una dirección válida y el momento en el que en las líneas de datos ( $DQ0$ -  $DQ7$ ) tenemos los datos de salida con valores estabilizados y, por consiguiente, válidos. En algunas casas comerciales se usa con frecuencia otra nomenclatura que es la que aparece en la columna paralela ( $t_{AVQV} = t_{ACC}$ ).

Para representar el significado de las distintas formas de onda que describen los procesos de lectura y escritura en una memoria EEPROM se usa en general un criterio como el que se ilustra en la *figura 7.23*. Cuando aparecen dos líneas paralelas significa que el dato o la dirección que representan está estable. Cuando hay líneas inclinadas bajando o subiendo de nivel significa que en ese intervalo temporal pueden haber cambio de alta a baja o de baja a alta. Cuando se superponen las líneas inclinadas descendentes y ascendentes es que no hay que preocuparse porque cualquier cambio está permitido en las líneas de entrada. En cambio, este mismo símbolo sobre una línea de salida significa que esta está cambiando y, por consiguiente, se desconoce su estado en ese intervalo. Finalmente, cuando hay una línea central en las salidas, significa que la puerta de tres estados correspondiente se encuentra en el estado de alta impedancia, desconectada funcionalmente del bus.

<i>Forma de Onda</i>	<i>Significado en las Entradas</i>	<i>Significado en las Salidas</i>
	Deben ser estacionarias	Valores estacionarios (salida válida)
	Pueden cambiar de H a L	Estará cambiando de H a L
	Pueden cambiar de L a H	Estará cambiando de L a H
	Situación irrelevante. Se permite cualquier cambio	Cambiando. Estado no conocido
	No se aplica	Estado de alta impedancia (línea central)

**Figura 7.23.** Significado de las distintas formas de onda que se usan para representar la evolución temporal de las líneas de datos y direcciones y las señales de control durante los procesos de lectura y grabación de EEPROMs.

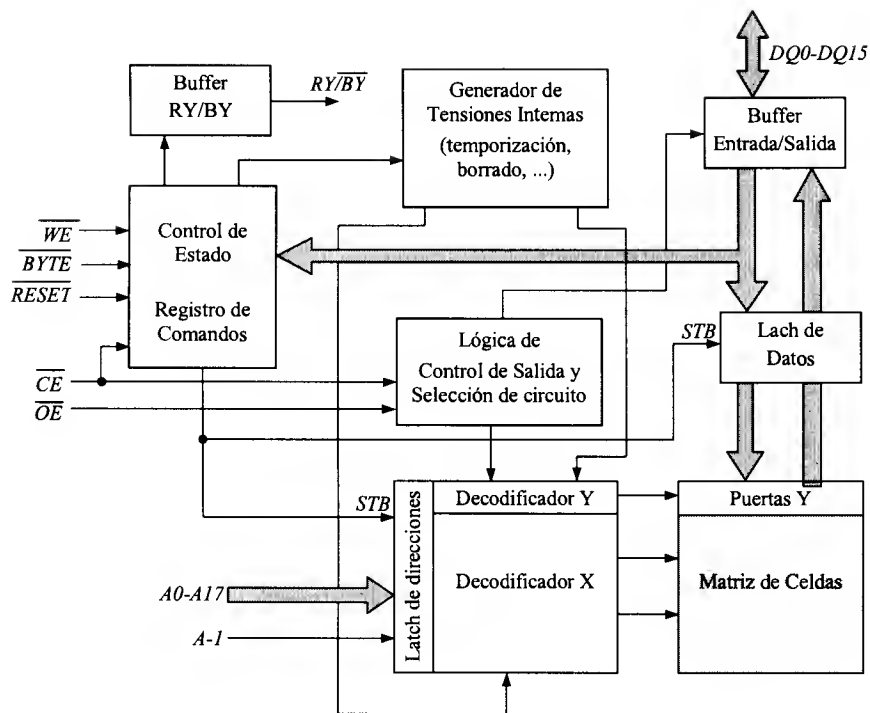
### 7.4.3. FLASH

La diferencia esencial en cuanto a organización interna entre las FLASH y las otras EEPROMs es que ahora se incluyen en el circuito (junto a las celdas de almacenamiento, el direccionamiento y los buffers de entrada y salida) un conjunto de circuitos adicionales para controlar internamente los programas de lectura y borrado por zonas a partir de un registro de “status” que contiene un conjunto de comandos considerados como instrucciones de entrada a un procesador interno (un autómata finito) que controla a los circuitos de borrado (global y por sectores) y programación. Durante los ciclos de escritura se registran internamente las direcciones y los datos necesarios para los programas de borrado y reprogramación que consisten, de hecho, en la ejecución de una secuencia de comandos.

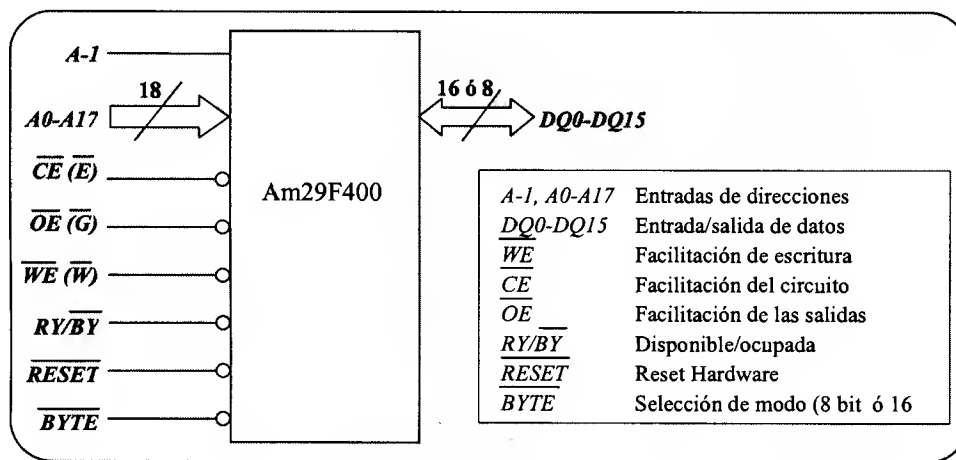
El circuito incluye también todas las transformaciones necesarias para obtener de una única fuente externa de continua todas las tensiones necesarias. Se consigue así integrar en el circuito las soluciones a la complejidad adicional de estas memorias para descargar al procesador externo, reduciendo al mínimo las señales externas de control y las exigencias de tiempo de CPU externa y facilitando el acoplo de la EEPROM FLASH a circuitos microprocesadores.

El diagrama de bloques de la *figura 7.24* muestra una versión simplificada de la estructura interna de la memoria FLASH Am29F400T (de AMD) de 4 Megabits ( $524,288 \times 8 / 262,144 \times 16$ ), CMOS a 5 voltios y posibilidad de borrado por sectores, con programas internos de borrado y programación. La *figura 7.25* muestra el símbolo lógico y el significado de la función realizada por sus terminales que, junto a los que ya conocemos ( $\overline{CE} = E$ ,  $\overline{OE} = G$  y  $\overline{WE} = W$ ) incluye otros nuevos ( $RY / \overline{BY}$ ,  $\overline{RESET}$  y  $\overline{BYTE}$ ).

El terminal  $RY / \overline{BY}$  se usa para informar al procesador externo si los programas internos están en curso o han terminado. Si su salida está en baja indica que el dispositivo está ocupado (“busy”) en una operación de borrado o reprogramación. Si la salida está en alta el dispositivo está preparado para aceptar cualquier operación nueva de lectura, escritura o borrado. El terminal  $\overline{RESET}$ , cuando está en baja durante más de  $500ns$ , resetea el procesador interno y se ignoran los valores de la señal  $RY / \overline{BY}$ . Finalmente, el terminal  $\overline{BYTE}$  selecciona la configuración de la memoria para 8 ó 16 bits.



**Figura 7.24.** Esquema simplificado de la estructura interna de la memoria FLASH de AMD, Am29F400T.



**Figura 7.25.** Símbolo lógico de la memoria FLASH de AMD, Am29F400T y significado de la función que realizan sus terminales.

Hay ejemplos más actuales<sup>(\*)</sup> como la memoria FLASH de bajo voltaje de ST, M59DR008E, de 8Mbit (512Kb x 16, Dual Bank, Page) alimentada con  $V_{DD}$  entre 1,65V y 2,2V para programar, borrar y leer y con organización en bloques y lectura por páginas. Así, cada bloque puede ser borrado y reprogramado sin afectar a los otros bloques. Este circuito acepta instrucciones para “leer/resetear”, “escribir en el registro de configuración”, “programar”, “borrar un bloque”, “borrar un banco”, “proteger un bloque”, etc., escritas en la memoria a través de una interfaz de comandos usando los cronogramas de escritura usuales en los microprocesadores.

Ya hemos comentado en varias ocasiones que desde el punto de vista pedagógico no es realmente esencial el tipo de circuito comercial que seleccionamos para ilustrar la evolución de la tecnología y el contacto con la realidad. Es decir, la comprensión y uso adecuado de las hojas de datos y las notas de aplicación de las distintas casas comerciales. Además, cuanto más activo es un sector (como es el caso de las EEPROMs de tipo FLASH) más rápido serán los cambios en las ofertas de circuitos concretos, por lo que no puede pretenderse en un texto de esta naturaleza más que “dar pistas” sobre prestaciones, funcionalidad y estructura interna de las memorias EEPROM.

## 7.5. PALS Y PLAS

Hemos visto la arquitectura de las PROMs, con la matriz AND fija y la matriz OR programable, y sus aplicaciones en el diseño lógico y como elemento de almacenamiento cuasi permanente de información, donde las soluciones eléctricamente reprogramables como las EEPROMs tipo FLASH les auguran un futuro prometedor.

Sin embargo, pensando sólo en su aplicación como módulo universal de diseño en lógica combinatorial, la arquitectura PROM tiene algunos problemas ya que la parte AND genera todos los términos mínimos de  $n$  variables ( $2^n$ ) y para añadir una nueva variable hay que duplicar el tamaño de la matriz. Además, en un número importante de situaciones de diseño muchos de esos términos mínimos no se usan. Por ejemplo, en aquellas situaciones en las que se quiere usar el mismo circuito programable para sintetizar funciones diferentes que necesitan una combinación no preestablecida de variables de entrada y variables de salida, o cuando se desea diseñar a la vez funciones

---

<sup>(\*)</sup> En el momento de la redacción de esta segunda edición.



combinacionales y secuenciales. En estas circunstancias es mejor disponer de un circuito que deje fija la parte OR y permita la programación de la parte AND. Así aparecieron los PAL (Programmable Arrays Logic) en 1978 como un dispositivo universal de diseño lógico introducido por Monolithic Memories, integrada después en AMD (Advanced Micro Devices). La *figura 7.26* muestra esta arquitectura para el caso sencillo de cuatro variables de entrada ( $x_3, x_2, x_1, x_0$ ) y cuatro funciones de salida ( $f_3, f_2, f_1, f_0$ ). Obsérvese que cada puerta OR sólo puede sumar un cierto número de términos mínimos (4 en el caso de la figura) lo que limita el repertorio de funciones posibles. Sin embargo, tiene la gran ventaja de no tener que duplicar el tamaño de la matriz cada vez que se necesite una nueva variable de entrada.

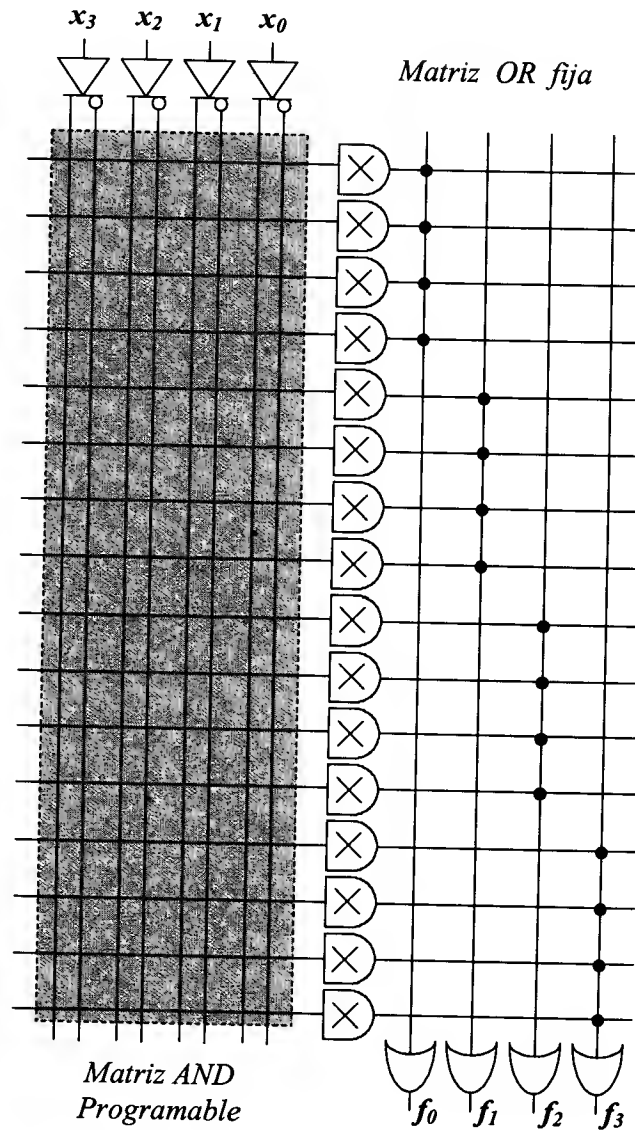
Hay circuitos PAL en tecnologías TTL y CMOS, programables por fusibles o eléctricamente, como las EEPROMs o las GAL (Generic Array Logic)<sup>(+)</sup>, y con distintas modificaciones de la arquitectura básica que incluyen la programación de la función de los terminales de salida, la realimentación y la adición de macroceldas con multiplexos y biestables también programables. Globalmente, al incluir biestables D ó J-K en las macroceldas de salida, las PAL pueden usarse tanto para diseño en lógica combinacional como para diseño secuencial. Sin embargo nosotros estudiaremos aquí básicamente la parte combinacional, dejando para más tarde (tema 13) la lógica secuencial programable. La razón obvia es que antes de usar los biestables y los registros en lógica programable, debemos estudiarlos de forma independiente. Sin embargo, la evolución de la tecnología electrónica nos obliga a veces a usar una idea intuitiva de biestable y registro, antes de estudiarlos en profundidad, para poder entender el comportamiento de ciertos circuitos inicialmente combinacionales, como las PROMs y las PALs, pero que en sus versiones más actuales ya incluyen elementos secuenciales.

Si lo que buscamos en el diseño combinacional es un modelo universal versátil, la arquitectura más plástica es la PLA (Programmable Logic Array) en la que tanto la matriz de puertas AND como la de puertas OR son programables, tal como se muestra en la *figura 7.27*. Evidentemente, la contrapartida de la versatilidad es la complejidad y el coste del circuito resultante. Por consiguiente,

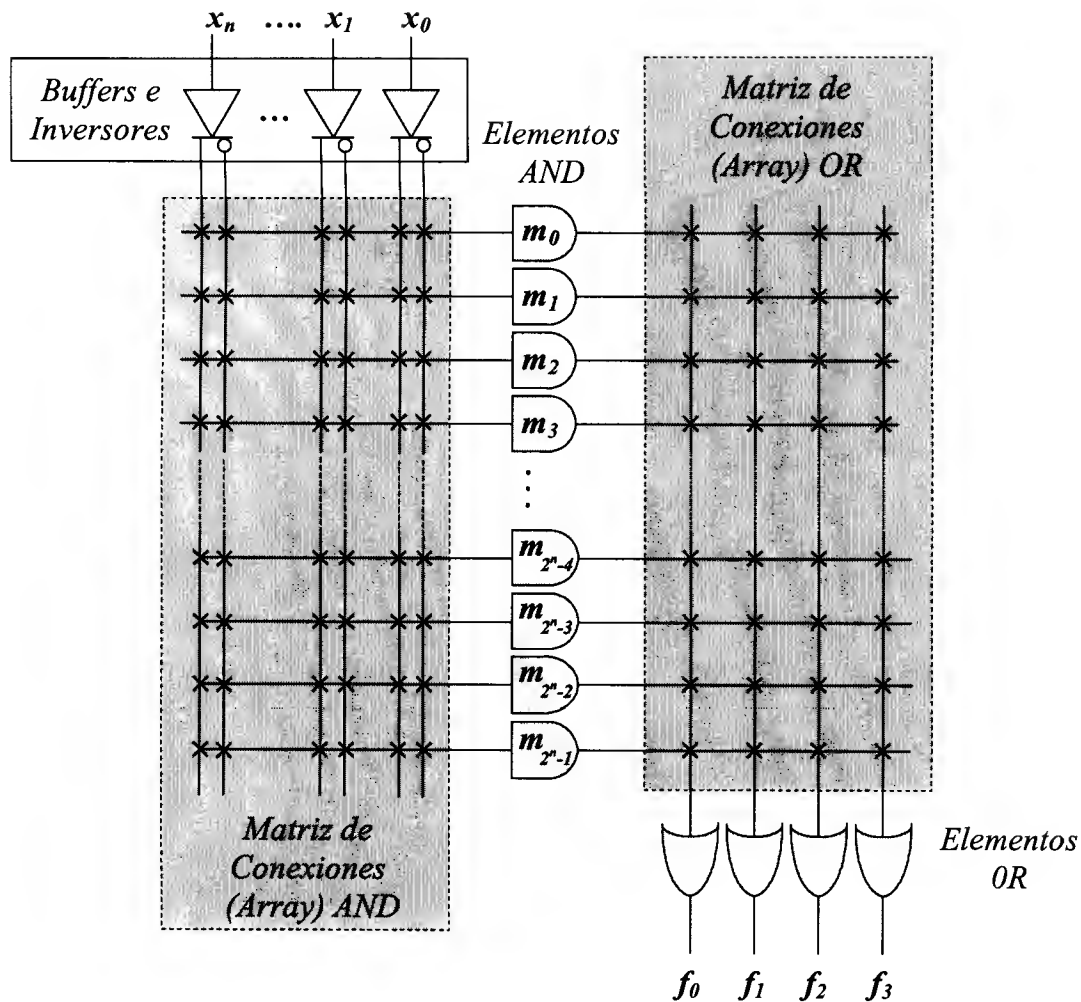
---

<sup>(+)</sup> Al igual que PAL fue un nombre acuñado inicialmente por Monolithic Memories y usado posteriormente de forma general, GAL es un nombre acuñado inicialmente por Lattice Semiconductors y usado también más tarde de forma general.

las PLAs son recomendables en aquellas situaciones de diseño en las que sean justificables las facilidades de programación que ofrecen.



**Figura 7.26.** Arquitectura de los circuitos programables PAL.



**Figura 7.27.** Arquitectura de los circuitos PLA.

Veamos con un ejemplo sencillo el uso de PLAs en diseño.

**Ejercicio:** Supongamos que disponemos de una PLA  $3 \times 4 \times 2$ . Es decir, una PLA con 3 líneas de entrada ( $x_0, x_1, x_2$ ), 4 productos o funciones AND y 2 líneas de salida ( $f_0, f_1$ ) y queremos programarla para que implemente las funciones:

$$f_0(x_0, x_1, x_2) = \sum m(3, 5, 6, 7)$$

$$f_1(x_0, x_1, x_2) = \sum m(0, 2, 4, 6)$$

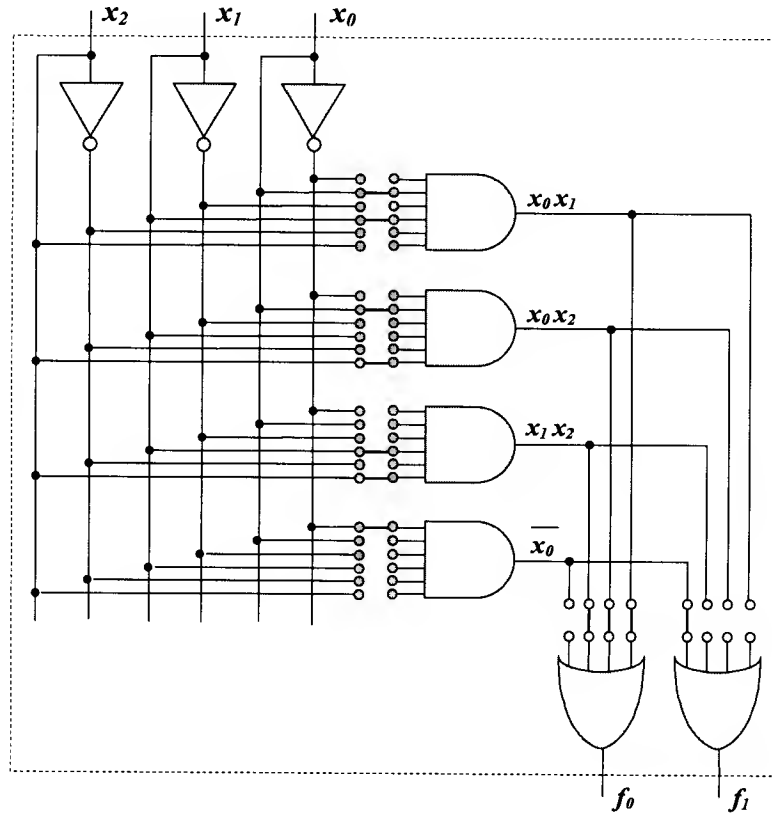
### Solución

Para implementar estas funciones lo primero que tenemos que hacer es simplificar las expresiones y minimizarlas. Así,

$$f_0(x_0, x_1, x_2) = \sum m(3, 5, 6, 7) = \bar{x}_2 x_1 x_0 + x_2 \bar{x}_1 x_0 + x_2 x_1 \bar{x}_0 + x_2 x_1 x_0 = x_1 x_0 + x_2 x_0 + x_2 x_1$$

$$f_1(x_0, x_1, x_2) = \sum m(0, 2, 4, 6) = \bar{x}_2 \bar{x}_1 \bar{x}_0 + \bar{x}_2 x_1 \bar{x}_0 + x_2 \bar{x}_1 x_0 + x_2 x_1 x_0 = \bar{x}_0$$

Con lo que la PAL 3\*4\*2 programada sería la que se muestra en la figura 7.28.



**Figura 7.28.** Síntesis con una PAL 3\*4\*2 de las funciones  $f_0(x_0, x_1, x_2) = \sum m(3, 5, 6, 7)$ ,  $f_1(x_0, x_1, x_2) = \sum m(0, 2, 4, 6)$ .

## 7.6. CONFIGURACIONES DE SALIDA



La parte de entrada de todos los circuitos PAL es esencialmente la misma y también se repite la componente central de su estructura interna (matriz AND programable y matriz OR fija). Sin embargo, hay distintos tipos de circuitos de salida que han ido complicándose para aumentar las funcionalidades de las PALs como elementos universales de diseño. Hay dos tipos de salidas: *combinacionales* y *secuenciales* y las versiones más complejas de las secuenciales (macrocelas) incluyen a las combinacionales como caso particular.

Las *salidas combinacionales* incluyen:

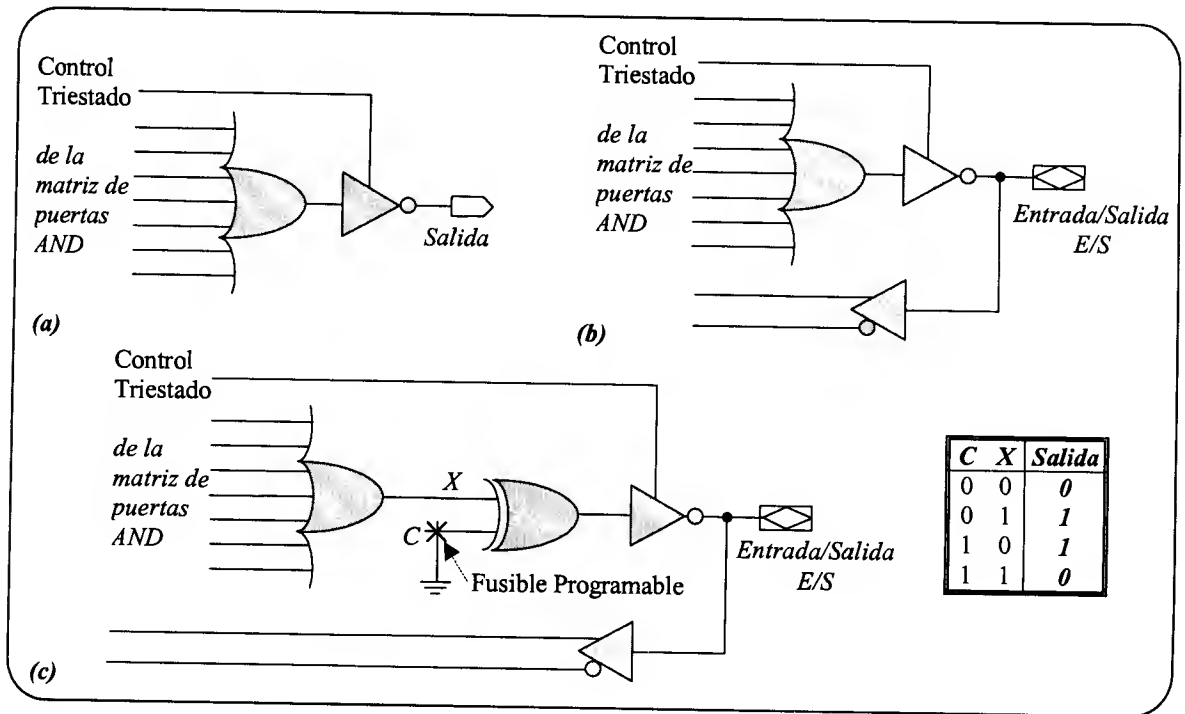
1. La posibilidad de sumar los términos mínimos directamente o a través de una puerta XOR para poder controlar la *polaridad* (salidas activas en baja o en alta).
2. La posibilidad de que algunas *salidas* puedan, cuando convenga, ser programadas también como *entradas*.
3. La posibilidad de *realimentación*.

Las *salidas secuenciales* son de dos tipos:

4. *Básicas*, con la inclusión de un circuito biestable para mantener el valor de la salida, retardarlo y realimentarlo cuando convenga.
5. *Macrocelas*, donde se incluyen también biestables junto a multiplexos para hacer más versátil la programación de las rutas seguidas por los datos.

La *figura 7.29* muestra los tipos básicos de la lógica de salida de una PAL. En *7.29(a)* se muestra el caso más básico. Simplemente, la puerta OR suma las salidas de las puertas AND y hay después un buffer de tres estados. El circuito que se muestra corresponde a la situación de “salida activa en nivel bajo”. Por eso aparece un buffer inversor. Además, como esta salida sólo puede realizar funciones de salida, se usa el símbolo  para representarla. Cuando una salida puede programarse para que opere como entrada se representa con el símbolo , como en *7.29(b)*. Aquí hay también posibilidad de realimentación. Si el buffer de tres estados está facilitado la salida puede realimentarse. Si está en alta impedancia lo que se “realimenta” es la señal externa, actuando entonces el terminal de salida como entrada. La señal de control del buffer de tres estados es programable a nivel individual, para cada una de las salidas y debe ser programada de acuerdo con la funcionalidad deseada para ese terminal. Es decir, si el terminal va a reusarse como entrada, el buffer debe estar en alta impedancia.

Si va a usarse como salida, el buffer debe estar facilitado y cuando se desee usar como entrada unas veces y como salida otras, el buffer deberá estar programado en concordancia. Finalmente, en 7.29(c) se ha incluido una puerta XOR entre la OR que suma los términos producto y el resto del circuito.



**Figura 29.** Distintas posibilidades de salida combinacional de un circuito PAL. **(a)** Salida básica (activa en baja). **(b)** Salida (también activa en baja) con posibilidad de ser programada como entrada y con realimentación. **(c)** Lo mismo que (b) pero incluyendo una puerta XOR para permitir programar la polaridad.

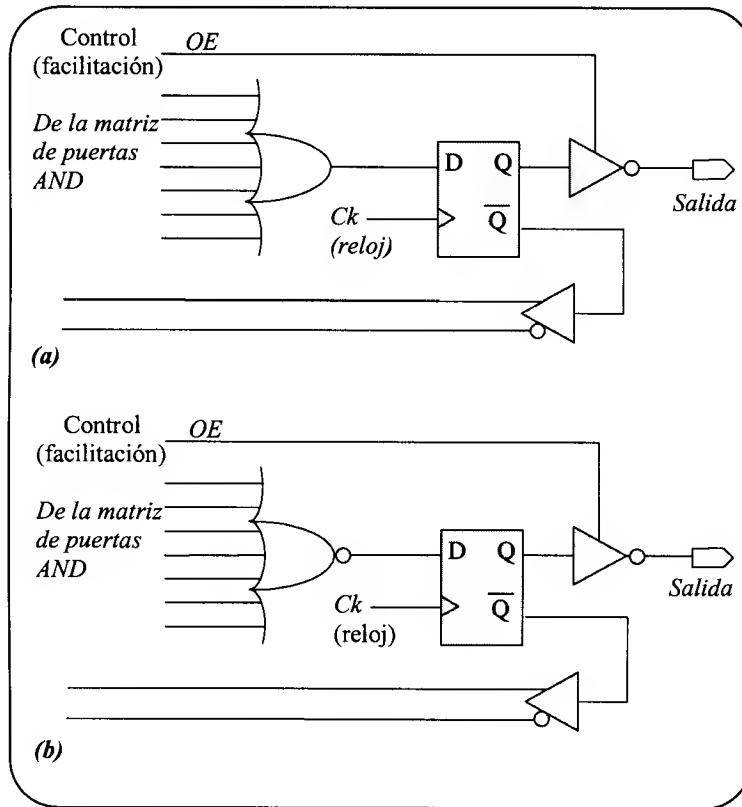
Cuando la entrada programable está en baja, la salida del XOR coincide con la salida de la puerta OR. En cambio, cuando está en alta el XOR invierte la salida de la puerta OR.

Todas las salidas secuenciales se basan en la inclusión de un biestable D<sup>(\*)</sup> entre la puerta OR y el resto de los circuitos de la salida combinacional (buffer y realimentación), tal como se muestra en la

<sup>(\*)</sup> Veremos en el tema siguiente este tipo de circuitos.

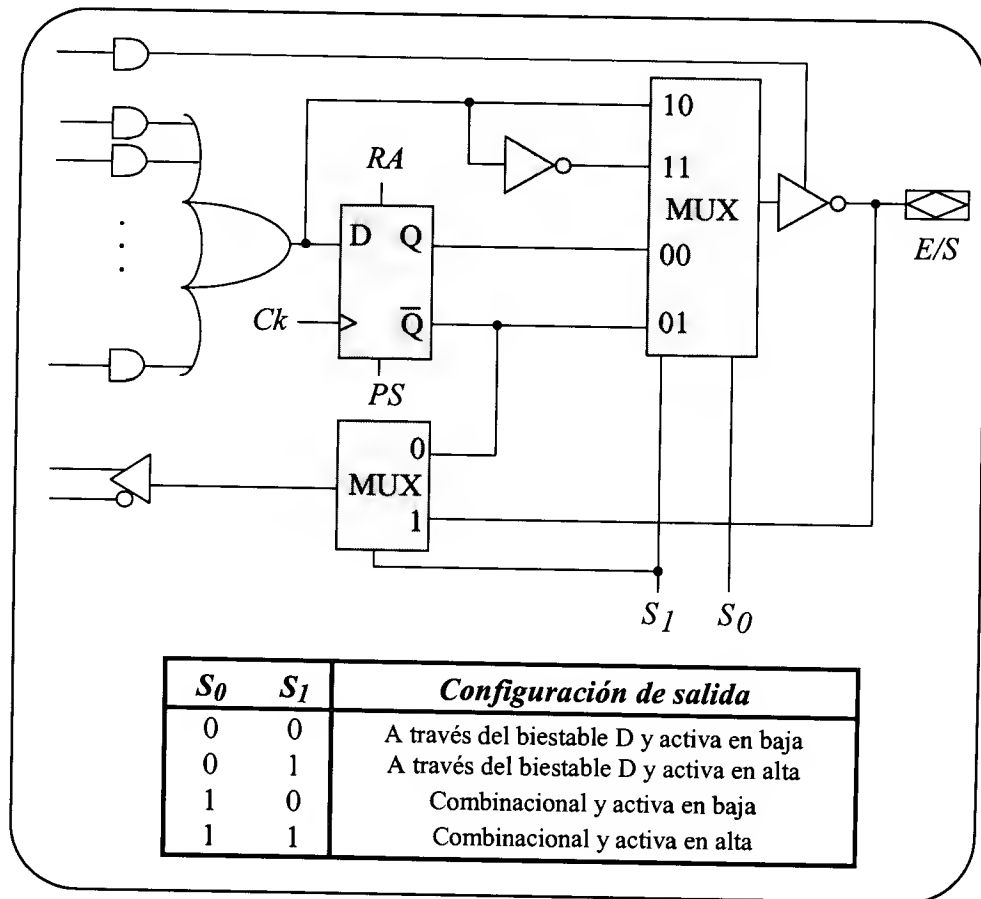
figura 7.30. Hay tres aspectos a destacar en este circuito:

1. La salida,  $Q(t)$ , coincide con el valor de la entrada (su “estado”) en el instante anterior,  $Q(t)=D(t-\Delta t)$ . Es decir, el biestable D es de hecho un retardo, lo que posibilita toda la lógica secuencial y, por consiguiente, el uso de la PAL como módulo universal de diseño de contadores, registros y autómatas finitos en general.
2. El circuito es síncrono. Es decir, todas las transiciones de datos desde  $D$  a  $Q$  y  $\bar{Q}$  ocurren a las subidas o bajadas de un pulso de reloj que es *común* a todas las salidas de la PAL.
3. La señal de control de los buffer es la misma para todas las líneas de salida.



**Figura 7.30.** Salidas de una PAL a través de un biestable D (registered output). **(a)** Activa en baja. **(b)** Activa en alta.

Finalmente, la configuración de salida más general es la macrocelda (“output logic macrocell, OLMC) que incluye un biestable D y dos multiplexos (uno de 4 a 1 y otro de 2 a 1) para permitir la programación de la ruta de datos y obtener, como casos particulares, las configuraciones combinacionales y secuenciales descritas anteriormente. La *figura 7.31* muestra el esquema de una macrocelda básica (PAL 22v10). Como es fácil comprobar, dependiendo del valor de las dos variables de control de los multiplexos ( $S_0$  y  $S_1$ ) la macrocelda se proyecta en los cuatro casos que se describen en el tabla incluida en la *figura 7.31*.



**Figura 7.31.** Esquema de una macrocelda de salida típica de circuitos PAL, PALCE y GAL.  $Ck$ =pulsos de reloj,  $RA$  = Reset asíncrono,  $PS$  =Preset sincrono..



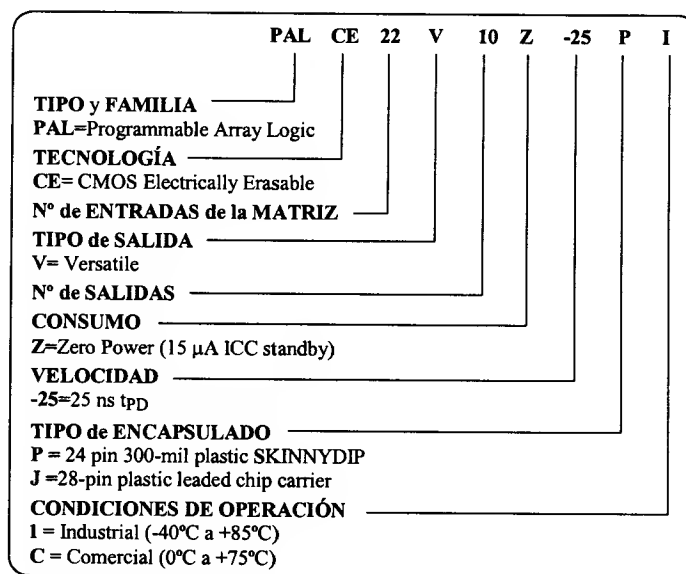
Así, para  $S_0 = 1$  y  $S_I = 0$  el multiplexo de 4 a 1 abre paso a la señal que procede directamente de la puerta OR, sin pasar por el biestable D, con lo que ignora la acción del biestable. Al mismo tiempo el multiplexo de 2 a 1 que está en la vía de realimentación ( $S_I = 1$ ) deja paso a la señal que procede de la salida del buffer, con lo que la macrocelda actúa como una configuración de salida combinacional. En este caso con salida activa en baja. En cambio, si  $S_I = 0$  y  $S_0 = 1$  tendremos una configuración secuencial activa en alta; es decir, a través del biestable D (ya que el multiplexo de 4 a 1 deja pasar la salida  $Q$  del biestable D) y tomando la realimentación de  $\bar{Q}$ . De forma análoga obtendríamos las otras dos configuraciones (secuencial con salida activa en baja y combinacional con salida activa en alta). Para facilitar la visualización de la operación de selección de entrada que realizan los circuitos multiplexores en su esquema hemos puesto las señales de entrada a la misma altura que las configuraciones que les facilitan el paso ( $D$  frente a 10 y  $\bar{D}$  frente a 11,  $Q$  frente a 00 y  $\bar{Q}$  frente a 01), y lo mismo en el multiplexo de 2 a 1 que controla la realimentación.

El biestable que incluye la macrocelda realiza funciones de almacenamiento (retardo) de los datos y de sincronización. Para ello incluye una *entrada de reloj* ( $Ck$ ), otra de *reset asíncrono* ( $RA$ ) y otra de *preset síncrono* ( $PS$ ). Para inicializar la PAL, si  $PS$  está en alta la salida  $Q$  se carga a alta en la siguiente transición de baja a alta de los pulsos de reloj. Cuando  $AR$  está en alta los biestables de salida se cargan a baja, independientemente de la situación en la que se encuentren en ese momento los pulsos del reloj. Obsérvese que estas líneas de control son términos productos que proceden de la parte NAND de la PAL.

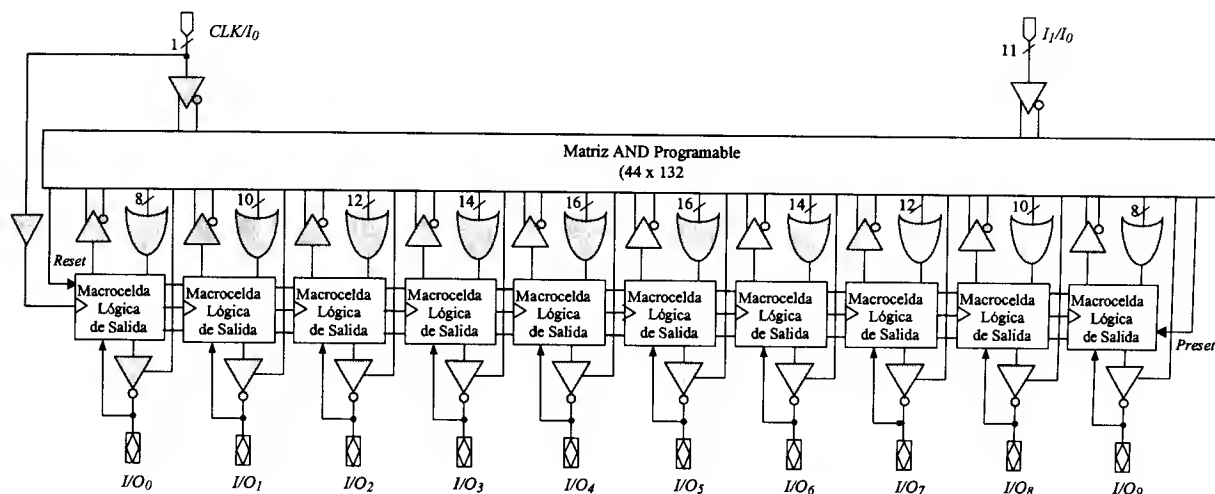
## 7.7. NOMENCLATURA Y EJEMPLO DE CIRCUITOS PAL

La evolución de las PAL ha caminado paralela a la de las PROMs, pasando de la tecnología bipolar y la programación por fusibles a la tecnología CMOS de bajo consumo y alta densidad de integración y el uso de celdas EEPROM borrables y reprogramables eléctricamente. Para marcar esa evolución las distintas casas comerciales han acuñado nombres diferentes para sus productos. Así, AMD llama PALCE a las familias de sus circuitos PAL de alta velocidad y bajo consumo en tecnología CMOS que son eléctricamente borrables (EEPROM). Al mismo tipo de circuito le llama GAL (Generic Array Logic) la compañía Lattice Semiconductor. Además son directamente reemplazables para el mismo número de terminales. La *figura 7.32* muestra la nomenclatura usada

para la familia PALCE y la *figura 7.33*, una versión simplificada del esquema de la PALCE 22V10Z-25.



**Figura 7.32.** Nomenclatura usada para la familia PALCE



**Figura 7.33.** Versión simplificada del esquema de la PALCE 22V10Z-25.

Al haber estudiado previamente la estructura interna de las macroceldas y la función que realizan las señales de control (*reset* asíncrono, *preset* síncrono y reloj), este circuito no es más que la repetición del esquema básico para cada una de las salidas OR. Obsérvese que las señales *AR* y *SP* proceden de términos productos en la matriz AND.

## 7.8. PROBLEMAS

Hemos introducido en este tema la lógica combinacional programable (PROM's, PLA's y PAL's), con lo que ya tenemos a nivel de diseño combinacional funciones lógicas universales en forma de sumas de términos mínimos que resumen las facilidades de los multiplexos, demultiplexos y decodificadores que estudiamos en el *tema* anterior. Por consiguiente podríamos repetir aquí todos los circuitos propuestos desde el comienzo del texto (ejemplos arbitrarios del *tema* primero, funciones aritmético-lógicas del *tema* quinto y funciones de ruta de datos del *tema* sexto).

Haremos una selección de este conjunto de ejercicios posibles recordando que la norma general de selección entre PROM, PAL y PLA es disponer de "*todo lo necesario*" y dejar sin usar "*lo menos posible*". Así, en situaciones de diseño que exigen funciones dependientes de muchas variables de entrada y pocas funciones de salida, es aconsejable usar PROM's. En cambio, para situaciones de diseño con pocas variables de entrada y muchas salidas, es conveniente usar PAL. Finalmente, conviene recordar que lo que vamos a proponer son ejercicios en el sentido académico del término.

**E.7.1.** Sintetizar las siguientes funciones:

$$f_1(X,Y,Z) = (\bar{X} \cdot Y + X \cdot \bar{Y}) \cdot Z + \bar{Z} \cdot (X \cdot Y + \bar{X} \cdot \bar{Y})$$

$$f_2(X,Y,Z) = \bar{X} + \bar{Y} \cdot Z + \bar{Z} \cdot X$$

$$f_3(X,Y,Z) = X \cdot \bar{Y} + Y \cdot \bar{Z} + \bar{X} \cdot \bar{Y} \cdot \bar{Z} + \bar{X} \cdot \bar{Y} \cdot Z + X \cdot Y \cdot Z$$

$$f_4(X,Y,Z) = (X + Y + \bar{Z}) + (X + \bar{Y} + \bar{Z}) + \bar{X} \cdot \bar{Y} \cdot Z + \bar{X} \cdot Y \cdot \bar{Z}$$

**E.7.2.** Sintetizar de la forma más "económica", usando PROM, PAL ó PLA, el sumador completo de un bit cuyas entradas son  $A_i$ ,  $B_i$  y  $C_i$  y las salidas son  $S_i$  y  $C_{i+1}$ .

Sintetizar a continuación el sumador paralelo para palabras de cuatro bits. Hay ahora nueve entradas ( $B_3, B_2, B_1, B_0; A_3, A_2, A_1, A_0; C_0$ ) y debe producir cinco

salidas ( $S_3, S_2, S_1, S_0; C_4$ ).

**E.7.3.** Sintetizar de la forma mas "económica", usando ROM, PAL ó PLA un circuito comparador para cuatro bits con salidas  $C(A_n > B_n)$  y  $E(A_n = B_n)$ .

**E.7.4.** Sintetizar un circuito sumador-restador para palabras de tres bits.

**E.7.5.** Sintetizar una "mini" unidad aritmético-lógica con sólo cuatro funciones para palabras de dos bits ( $A_0, A_1; B_0, B_1$ ). Necesitará por tanto dos variables de control ( $S_0, S_1$ ) y una adicional para la selección de modo ( $M=0 \Rightarrow$  operación aritmética y  $M=1 \Rightarrow$  operación lógica).

Tenemos entonces un circuito combinacional con siete variables de entrada (arrastre;  $A_0, A_1; B_0, B_1; S_0, S_1, M$ ) y dos variables de salida ( $F_0, F_1$ ) que dependiendo del valor de  $M$  representarán las funciones aritméticas ó lógicas que se muestran en la tabla adjunta

SELECCIÓN		FUNCIONES LÓGICAS	FUNCIONES ARITMÉTICAS
$S_1$	$S_0$	$M=1$	$M=0$
0	0	$F_0 = A \cdot B$	$F_0 = \bar{A}\bar{B}$ plus $A$
0	1	$F_1 = A + B$	$F_1 = (A + B)$
1	0	$F_2 = \bar{A}\bar{B} + \bar{A}B$	$F_2 = A$ plus $B$
1	1	$F_3 = \bar{A}\bar{B} + A B$	$F_3 = A$ minus $B$ minus 1

**E.7.6.** Diseñar un circuito que sintetice la función:

$$f(X, Y, Z, U, V) = \bar{X}\bar{Y}\bar{Z}\bar{U}V + \bar{X}\bar{Y}\bar{Z}U\bar{V} + X\bar{Y}ZU\bar{V} + X\bar{Y}\bar{Z}U\bar{V} + XYZUV + XY\bar{Z}\bar{U}\bar{V}$$

**E.7.7.** Diseñar los siguientes circuitos convertidores de código.

a) 4 a 4, BCD a Exceso-3.

b) 4 a 10, BCD a decimal.

c) Decimal a BCD.



---

# Preparación de la Evaluación

---

## Objetivos

- 1: *Modelo formal subyacente a las PLDs*
- 2: *PROMs, EEPROMs y FLASH*
- 3: *PALs (GALs) y PLAs*

## 1. Modelo formal subyacente a las PLDs

1.1. Hemos visto al comienzo del tema cómo la representación de una función lógica en forma normal disyuntiva (como suma de productos) daba origen a la lógica combinacional programable al ofrecernos una estructura universal en la que, modificando el valor de un conjunto de parámetros, se podían obtener todas y cada una de las funciones lógicas de  $n$  variables.

¿Podría revisar esa introducción e intentar formular otra dual?. Es decir, ¿cómo serían las funciones universales ( $y_0, y_1$ ) de dos variables ( $x_0, x_1$ ) representadas en términos de maxterms (productos de sumas)?. ¿Cuáles serían las equivalentes en esta representación a las arquitecturas PROM, PAL y PLA?.

1.2. Compruebe su nivel de conocimiento sobre la representación simplificada de puertas AND y OR múltiples y sobre la distinción entre conexiones permanentes, programables intactas y programables fundidas en PLDs.

a) Pase a simbología propia de PLDs las funciones siguientes

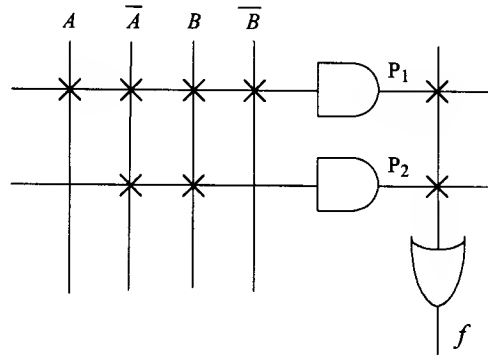
$$f_1(X,Y,Z) = (\overline{X} \cdot Y + X \cdot \overline{Y}) \cdot Z + \overline{Z} \cdot (X \cdot Y + \overline{X} \cdot \overline{Y})$$

$$f_2(X,Y,Z) = \overline{X} + \overline{Y} \cdot Z + \overline{Z} \cdot X$$

$$f_3(X,Y,Z) = X \cdot \bar{Y} + Y \cdot \bar{Z} + \bar{X} \cdot \bar{Y} \cdot \bar{Z} + \bar{X} \cdot \bar{Y} \cdot Z + X \cdot Y \cdot Z$$

$$f_4(X,Y,Z) = (X + Y + \bar{Z}) + (X + \bar{Y} + \bar{Z}) + \bar{X} \cdot \bar{Y} \cdot Z + \bar{X} \cdot Y \cdot \bar{Z}$$

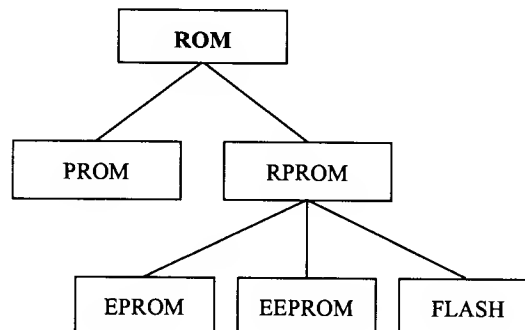
- b) *Realice la conversión inversa. Es decir, pase el siguiente circuito de notación PLD a notación convencional*



## 2. ROMs, PROMs, EEPROMs

2.1. *Todas las ROM son, esencialmente, funciones múltiples (m funciones de salida de n variables de entrada). Hay, sin embargo, diferencias en la forma de programación (por fusible, por luz ultravioleta, eléctricamente -por pulsos de tensión-, por conexiones individuales, globalmente, etc.) y en los circuitos adicionales que se le añaden para facilitar el direccionamiento o el proceso de grabación y borrado, por ejemplo.*

*¿Podría describir, de acuerdo con estos dos criterios (forma de programación y circuitos adicionales), las distintas versiones de memorias ROM?:*



**2.2. Las aplicaciones más generales de las PROM son:**

- a)** *La síntesis de funciones lógicas*
- b)** *El almacenamiento relativamente pasivo de la información que necesita un procesador (programas, tablas, etc...).*

*En el ejercicio E.7.1 del texto de problemas ilustramos la primera de estas líneas de aplicaciones, como módulos universales de diseño. Revise este problema e intente resolver los siguientes, que son análogos:*

*Sintetizar con PROMs las funciones:*

$$f_1(x, y, z) = \Sigma m(1, 3, 5, 7)$$

$$f_2(x, y, z) = \Sigma m(0, 2, 4, 6)$$

**2.3. En el ejercicio E.7.2 se sintetiza un sumador binario completo para un bit y otro paralelo para 4 bits usando una PROM. Revise este problema y trate de resolver ahora estos otros que son análogos:**

- a)** *Sintetizar con PROMs un sumador paralelo con acarreo adelantado para palabras de 4 bits.*
- b)** *Sintetizar con PROMs un sumador de palabras de 2 bits en complemento a 1 (C-1) incluyendo el detector de rebose. (Usar como punto de partida el ejercicio 2.7 del apartado de preparación de la evaluación del tema 5. ¿Qué problemas le encuentra a este diseño para pasar de ser un ejercicio académico a ser una opción real de diseño?. ¿Considera que se aprovecha adecuadamente la parte AND?. ¿Qué alternativas propondría dentro de la lógica programable?.*

**2.4. ¿Qué modificaciones tendría que incluir en las soluciones del ejercicio anterior para convertirlo en un sumador en complemento a 2 (C-2)?. Le sugerimos que visite los ejercicios de evaluación 1.3 y 2.8 del tema 5.**

**2.5. Entre los criterios de clasificación de las PROMs (funcionalidad, volatilidad, direccionamiento y mecanismo físico soporte) es esencial el último, que facilita la posibilidad de reprogramación y la lectura/escritura, junto con la permanencia de los datos. La estructura física esencial es la de “canal enterrado” y transistor FAMOS. ¿Podría describirla?. ¿Por qué el cambio**

controlado en el valor del umbral,  $V_T$ , es tan importante?.

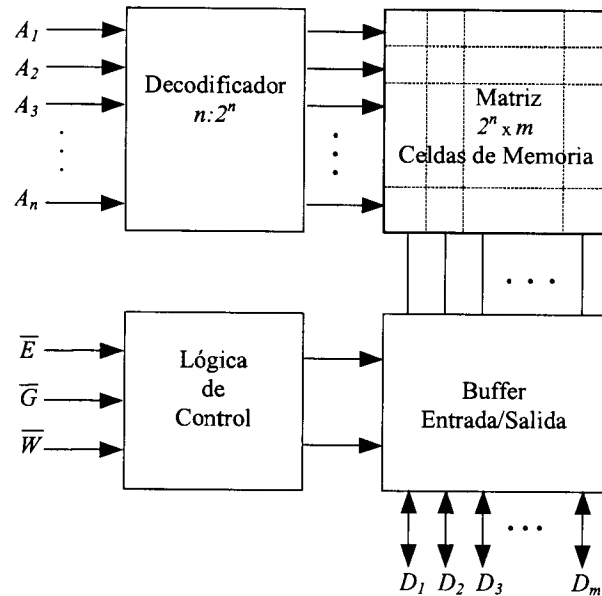
- 2.6. Describa cualitativamente el movimiento de carga entre el canal enterrado y el volumen del semiconductor durante las operaciones de grabación y borrado de una celda FLASH.
- 2.7. La figura adjunta muestra la arquitectura de una EEPROM que incluye, junto a las matrices AND (decodificador  $n:2^n$  completo) y OR (matriz de almacenamiento con  $2^n$  líneas de entrada y  $m$  líneas de salida), los amplificadores de tres estados (input/output buffers), las líneas bidireccionales de entrada/salida,  $D_1, \dots, D_m$  y la lógica de control:

$\overline{E}$  = Selección de circuito (chip enable)

$\overline{G}$  = Selección de salida (output enable) para acoplo con el bus

$\overline{W}$  = Control de la escritura de datos (write enable)

¿Podría explicar su funcionamiento tanto para grabación como para lectura?. Recuerde la descripción que se hace en el tema sobre la EEPROM M28C16A.



- 2.8. ¿Qué distingue a una EEPROM tipo FLASH de las otras EPROM vistas anteriormente?: a) A nivel de procesos físicos soporte. b) A nivel de comportamiento externo. c) A nivel de nuevos circuitos integrados en el chip.



### 3. Arquitecturas PAL y PLA

3.1. Hemos visto la arquitectura de las PROMs y algunas de sus aplicaciones en diseño lógico como elemento de almacenamiento cuasi permanente de información. Sin embargo, pensando sólo en su aplicación para la síntesis de funciones en lógica combinacional quedan en muchos casos desaprovechadas. ¿Podría explicar por qué?. ¿Cómo enlazan estas deficiencias de las PROMs con las arquitecturas PAL?

3.2. La figura 7.26 muestra el esquema de una PAL de 4 entradas y 4 salidas. Haga el estudio comparativo del coste de implementación con PALs de este tipo y con PROMs de 4 líneas de entrada y 4 de salidas para la síntesis de las siguientes funciones:

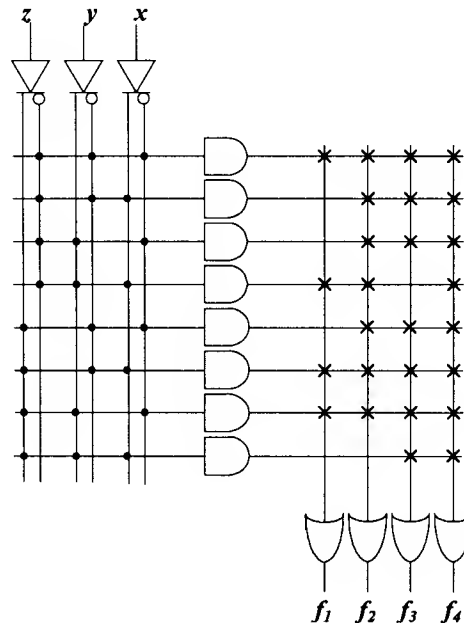
$$f_1(x, y, z) = (\bar{x} \cdot y + x \cdot \bar{y}) \cdot z + \bar{z} \cdot (x \cdot y + \bar{x} \cdot \bar{y})$$

$$f_2(x, y, z) = \bar{x} + \bar{y} \cdot z + \bar{z} \cdot x$$

$$f_3(x, y, z) = x \cdot \bar{y} + y \cdot \bar{z} + \bar{x} \cdot \bar{y} \cdot \bar{z} + \bar{x} \cdot \bar{y} \cdot z + x \cdot y \cdot z$$

$$f_4(x, y, z) = (x + y + \bar{z}) + (x + \bar{y} + \bar{z}) + \bar{x} \cdot \bar{y} \cdot z + \bar{x} \cdot y \cdot \bar{z}$$

3.3. ¿Qué relaciones encuentra entre el circuito de la figura adjunta y el enunciado del ejercicio anterior?



3.4. Hay situaciones en las que en una PAL nos sobran entradas y nos faltan términos mínimos para la síntesis de las funciones necesarias en una determinada aplicación. En estos casos se puede usar alguna salida como variable intermedia que se realimenta como una nueva entrada y contribuye así a la síntesis de las funciones de salida. Veámoslo en un ejemplo.

Se nos dice que sólo disponemos de una PAL de 4 entradas y 4 salidas (como la del ejercicio 3.2) y una puerta OR adicional para sintetizar las siguientes funciones:

$$f_1 = \sum m(0,3,5,6)$$

$$f_2 = \sum m(0,1,2,3,4)$$

$$f_3 = \sum m(0,1,2,4,5,7)$$

$$f_4 = \sum m(0,1,2,3,4,5,7)$$

Intente comprobar que esto es posible proponiendo un valor para la variable intermedia ( $u$ ) y un esquema de conexiones.

### **Solución**

Si tras reflexionar sobre el enunciado de este ejercicio no encuentra una solución satisfactoria, compruebe que el circuito de la figura resuelve el problema.

Obsérvese que sólo tenemos 4 términos mínimos para cada una de las 4 salidas, de forma que una salida la usaríamos para obtener  $f_1$ , y otras dos para obtener  $f_2$  y  $f_3$ . Como  $f_2$  posee 5 términos mínimos  $f_3$  posee 6 términos mínimos nos harían falta 2 salidas para cada una. Sin embargo, como 4 de estos términos mínimos (0, 1, 2, 4) son compartidos, podemos usar una variable intermedia  $u$  para sintetizar estas dos funciones:

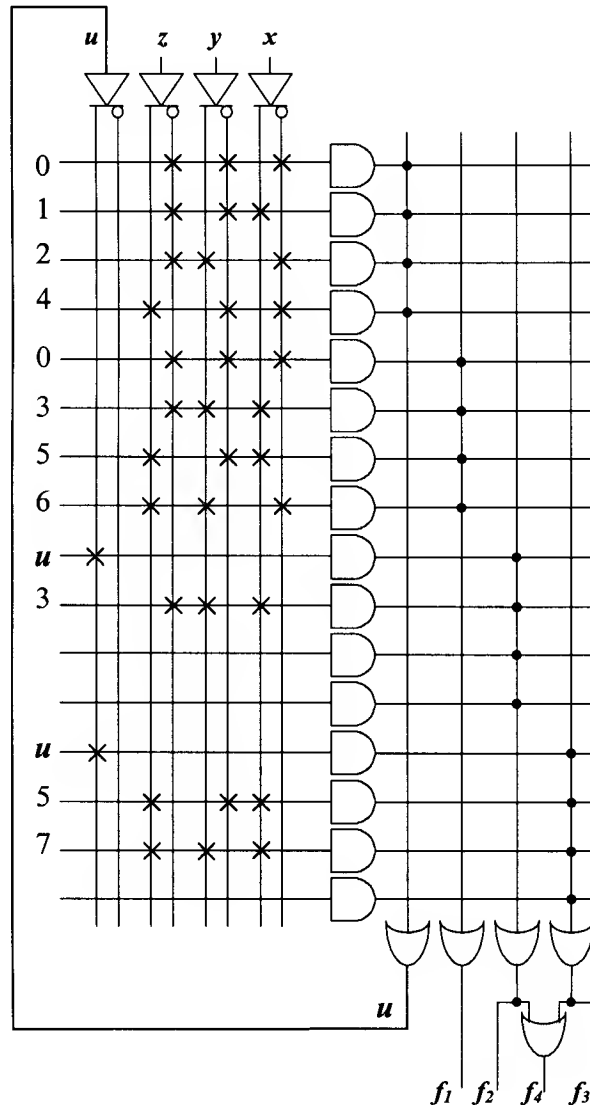
$$f_2 = u + \sum m(3)$$

$$f_3 = u + \sum m(5,7)$$

Finalmente  $f_4$  puede obtenerse de la unión  $f_2$  y  $f_3$ . Evidentemente, esta solución no es la única posible y no tiene por qué coincidir con otras que obtenga el lector.

El propósito de este ejercicio es de naturaleza puramente pedagógica y pretende ilustrar: (1) El uso de la realimentación de variables intermedias para transportar términos mínimos compartidos y

(2) el uso de puertas adicionales para aumentar la funcionalidad de los PLDs. Veremos en el tema 13 cómo estas estrategias de usar lógica adicional y de compartir y redistribuir términos mínimos han sido incorporadas en los nuevos diseños de CPLDs.



**3.5.** *Implemente un sumador/restador de palabras de 2 bits con PALs y con PROMs y haga el estudio comparativo del coste de ambas implementaciones.*

**3.6.** *La configuración de salida más general en las PALCE y las GAL es la macrocelda que permite*

configurarse en cuatro modos en función del valor de dos variables de control ( $S_0, S_1$ ). ¿Podría dibujar los cuatro circuitos concretos a los que da lugar la macrocelda de la figura 7.31 cuando ( $S_0, S_1$ ) pasan por las configuraciones posibles?.

- 3.7. En el ejercicio E.7.5 se propone la síntesis de una mini-ALU con sólo cuatro funciones para palabras de 2 bits ( $A_0 A_1, B_0, B_1$ ) usando 2 variables de control ( $S_0, S_1$ ) y una selección de modo ( $M=0$  para operación aritmética y  $M=1$  para operación lógica). Repasar este ejercicio para extraer las tablas de verdad y sintetizarlas ahora usando la PLA  $x*y*z$  (entradas, productos, salidas) que considere óptima, por ser la mínima necesaria y suficiente para las exigencias computacionales del problema. Comparar el resultado de este ejercicio con el del texto de problemas E.7.5.
- 3.8. En el ejercicio E.7.7 del texto de problemas se resuelve la síntesis de distintos circuitos conversores de código usando PLDs. Revise ese ejercicio y rellene la siguiente tabla:

Conversiones	Mejor con ROM, PAL ó PLA	¿Por qué?
4 a 4, BCD a Exceso a 3		
4 a 10, BCD a Decimal		
Decimal a BCD		



Con este tema terminamos el estudio de la lógica combinacional, incluyendo los aspectos básicos de los PLDs. Ya hemos comentado que las versiones más avanzadas de estos circuitos incluyen elementos de electrónica secuencial (biestables, registros, etc...). Por esa razón hemos segmentado su estudio en dos partes. La primera, que sólo usa lógica combinacional la hemos visto aquí. La siguiente la veremos al final de la *lógica secuencial convencional*, en el tema 13.

## REFERENCIAS BIBLIOGRÁFICAS

1. Advanced Micro Devices (AMD): “*PAL Device Data Book*”, 1992.
2. AMD: “*FLASH Memory Products Data Book*”, 1995.
3. Cypress Semiconductor: “*High Performance Data Book*”, 1995.
4. Floyd, T.L.: “*Fundamentos de Sistemas Digitales*”. Prentice Hall, 2000.
5. Rabaey, J. M.: “*Digital Integrated Circuits*”. Prentice-Hall. 1996.
6. Sandige, R.S.: “*Modern Digital Design*”. McGraw-Hill. 1990.
7. STMicroelectronics (<http://www.st.com>). 1998.
8. Sze, S.M.: “*Physics of Semiconductor Devices*” J. Wiley & Sons, 1981.
9. Weste, N.H.E., Eshraghian, K.: “*Principles of CMOS VLSI Design*”. Addison-Wesley. 1994.



TEMA

8

---

## ***Exigencias Computacionales de la Lógica Secuencial: Circuitos Biestables***

---

### **+ CONTEXTO**

*La primera parte del texto se ha dedicado al estudio de los circuitos combinacionales. Empezamos estudiando las exigencias computacionales del diseño en lógica combinacional. Vimos después las bases electrónicas necesarias para la síntesis de un conjunto completo de operadores. A continuación estudiamos algunas de las familias lógicas integradas y pasamos al nivel MSI estudiando las funciones necesarias para el cálculo aritmético-lógico y para controlar el tráfico de datos en un sistema de cálculo digital. Finalmente, en el tema anterior estudiamos la lógica combinacional programable. El aspecto distintivo de todos estos circuitos que hemos estudiado en la primera parte del texto es el carácter instantáneo de la relación entre las entradas y las salidas. Es decir, un circuito combinacional ideal produce a la salida una señal que sólo depende del valor que poseen las entradas en ese mismo instante. Es cierto que existen retardos de propagación que definen además la velocidad de conmutación de las distintas familias lógicas pero esos retardos no son*

*intrínsecos al modelo formal sino consecuencia del proceso físico de fabricación y, en definitiva, una limitación que la realización física impone sobre el modelo formal subyacente (el álgebra de Boole).*

*Ahora, en esta segunda parte del texto, vamos a introducir intencionalmente el retardo para poder diseñar circuitos secuenciales, con “memoria”, en los que la respuesta en un determinado instante,  $y(t)$ , no depende sólo del valor de sus entradas en ese instante,  $x(t)$ , sino que depende también de las entradas y respuestas en instantes anteriores,  $x(t-\Delta t)$ ,  $y(t-\Delta t)$ . Para que esto sea posible necesitamos, formal y físicamente, un tipo especial de circuito (al que llamamos biestable) que sea capaz de almacenar durante un cierto intervalo de tiempo,  $\Delta t$ , el valor de una señal digital. Como una señal digital tiene dos valores lógicos posibles, (“0”, “1”), necesitamos circuitos electrónicos capaces de acomodar (“almacenar”) en su estructura de forma permanente dos estados eléctricos distintos (0 y 5 voltios, por ejemplo). Por eso se llaman biestables (R-S, J-K, T y D), y con ellos iniciamos el estudio de la electrónica secuencial, dedicándole la mayor parte de este tema.*

*Una vez que conozcamos el funcionamiento de los biestables y la teoría de autómatas ya podremos sintetizar circuitos contadores, registros de desplazamiento, temporizadores y relojes, memorias RAM, CAM y FIFO, y lógica secuencia programable (CPLDs y FPGAs). Es decir, el resto de las funciones necesarias para el diseño digital en general y para el diseño de computadores en particular.*

*Este contexto que introducimos aquí está compartido por el resto de los temas (9 al 13) entre los que hemos distribuido las distintas funciones secuenciales de acuerdo con el esquema propuesto en el plan general del texto.*

## **++ CONOCIMIENTO PREVIO NECESARIO**

*En el diseño secuencial usamos finalmente conceptos y circuitos que ya conocemos. Es decir, en los biestables lo nuevo es la arquitectura, el esquema de conexión, pero al final terminamos diseñando con sólo puertas NOR ó usando sólo puertas NAND.*

*Análogamente, en los autómatas finitos, lo nuevo es la arquitectura (retardar y realimentar) y los aspectos de representación formal (diagramas de transición de estados, tablas, ...) pero al final las funciones encargadas de la transición de estados y de la producción de salidas son, de nuevo, funciones combinacionales convencionales, como las que hemos estudiado en los temas 1, 5, 6 y 7.*

*Por consiguiente, todo el conocimiento previo necesario está en esos temas.*

### **+++ OBJETIVOS DEL TEMA**

*El considerar la existencia de elementos de “memoria” en el análisis y la síntesis de circuitos digitales nos obliga a dos cosas:*

- I. Ampliar el modelo matemático soporte de la electrónica digital pasando del álgebra de Boole (suficiente para los circuitos combinacionales) a la teoría de autómatas finitos. Esto conlleva, además, ampliar las arquitecturas incluyendo la realimentación que cierra uno o varios lazos en los que se incluyen funciones combinacionales y elementos de memoria.*
- II. Ampliar el conjunto de operadores básicos. Antes nos bastaba con AND, OR, NOT ó NAND y ahora necesitamos añadir los biestables cuya función fundamental es el retardo: Un circuito que en el instante  $t + \Delta t$  nos da a su salida la entrada que tenía en  $t$  ó una función de la misma.*

*Por consiguiente, los objetivos generales del tema son:*

- Objetivo 1:** *Comprender aspectos conceptuales básicos en teoría de autómatas finitos, incluyendo el manejo del tiempo en diseño digital.*
- Objetivo 2:** *Entender con cierta profundidad el comportamiento de los distintos tipos de biestables, tanto por su configuración de disparo (D, T, R-S, J-K) como por su estructura interna (niveles, disparados por flancos, tipo “maestro-esclavo”, etc...).*

### **++++ GUÍA DE ESTUDIO**

*La descripción de los biestables (o flip-flops) es muy estándar y puede encontrarse con pequeñas diferencias en un gran número de textos de Electrónica Digital.*

*Nuestra estrategia al preparar el material ha sido de naturaleza incremental. Empezamos con la estructura más simple (los llamados “latch” R-S)<sup>(\*)</sup>, vemos los problemas que plantea, sugerimos*

---

<sup>(\*)</sup> Preferimos usar “latch” en vez de su traducción (“cerrojo”) y explicar su significado en el apartado correspondiente (biestables R-S).



*un procedimiento de solución que, obviamente, complica algo el circuito etc... Finalmente, complementamos esta descripción con la visión funcional externa, usando expresiones lógicas, tablas de verdad y diagramas de transición de estados.*

*La introducción a la teoría de autómatas hemos procurado hacerla intuitiva (ejemplo del bolígrafo) y no recargarla demasiado de aspectos formales. Su contenido también es convencional y fácil de seguir por distintos textos. Sólo queremos comentar un punto para guiar el estudio. No hay que olvidar que el material de este texto está asociado a una asignatura que no está aislada en el desierto, sino que forma parte de un plan de estudios en el que se estudia también lógica y teoría de autómatas, junto a otras materias de programación y arquitectura de computadores. Un número importante de los conceptos que se estudian aquí, están descritos también en términos parecidos en esas otras materias, por lo que aconsejamos una visión comprehensiva del tema.*



---

## Contenido del Tema

---

### 8.1. INTRODUCCIÓN A LOS AUTÓMATAS FINITOS: CONCEPTO DE ESTADO

El concepto distintivo de la electrónica secuencial y, por consiguiente, del modelo formal subyacente (la teoría de autómatas), es el concepto de *estado* interno de un circuito. Veámoslo sobre un ejemplo sencillo. Consideremos como sistema un bolígrafo de tipo pulsador en el que al pulsar sale la punta si estaba dentro y entra si estaba fuera. Para describir el comportamiento del sistema bolígrafo necesitamos un espacio de entradas con una variable,  $x$ , que toma dos valores:

$$x = x_1 = 1 = \text{pulsar} \qquad x = x_2 = 0 = \text{no pulsar}$$

El bolígrafo puede estar en dos estados,  $S$ :

$$S = S_1 = \text{punta dentro} \qquad S = S_2 = \text{punta fuera}$$

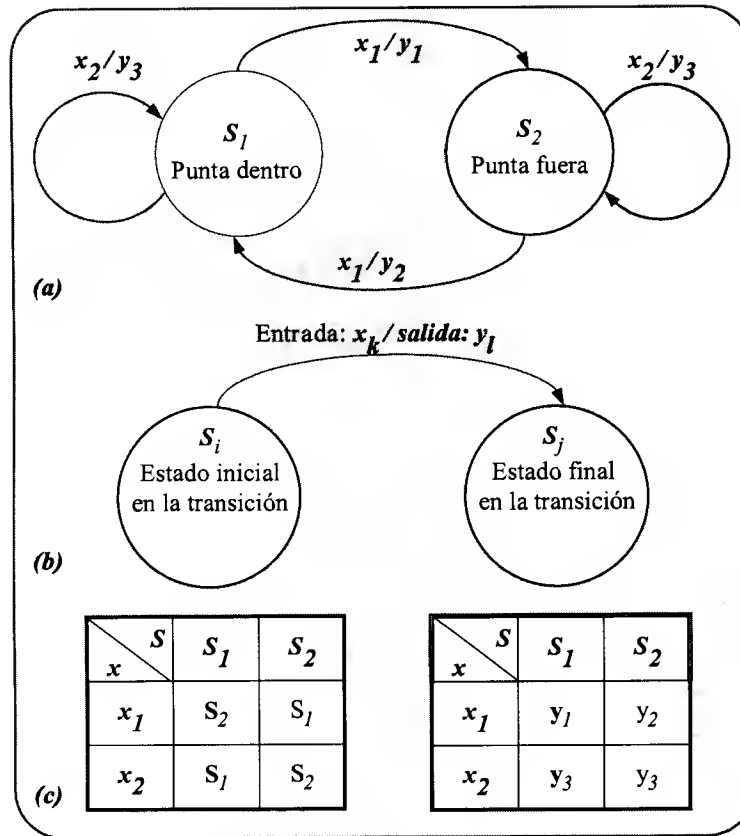
y las respuestas posibles al pulsar (espacio de salidas,  $y$ ) serían:

$$y = y_1 = \text{sale la punta} \qquad y = y_2 = \text{entra la punta} \qquad y = y_3 = \text{no se mueve}$$

Como consecuencia de existir dos estados internos distintos, una misma entrada puede producir dos salidas diferentes dependiendo del estado en el que se encuentre el sistema. Es decir, en general, la respuesta es función de la entrada y del estado y esto se ilustra en el diagrama de transición entre estados de la *figura 8.1.a*.

Esta descripción es completa e independiente del tiempo, de manera que al aplicarla de forma sucesiva para una determinada ristra de entradas de longitud siete, por ejemplo:  $x_1 x_2 x_2 x_1 x_2 x_1 x_1$ , y suponiendo que el estado inicial era  $S_1$ , por ejemplo, la ristra de salidas será:  $y_1 y_3 y_3 y_2 y_3 y_1 y_2$ .

Esta forma de representación sigue la ley de la *figura 8.1.b*. Los estados se encierran en círculos y entre dos estados cualesquiera puede haber siempre una rama que los enlaza sobre la que hay dos símbolos. Uno representa la entrada y otro la salida.



**Figura 8.1.** (a) Diagrama de transición de estados del sistema-bolígrafo. (b) Ley general de representación de las transiciones entre estados indicando en el arco que los une la entrada que provoca la transición y la salida que se produce como consecuencia de esa entrada y de esa transición. (c) Tablas de producción de salidas y nuevos estados.

Veamos pues que en la descripción de un circuito secuencial hacen falta las siguientes entidades:

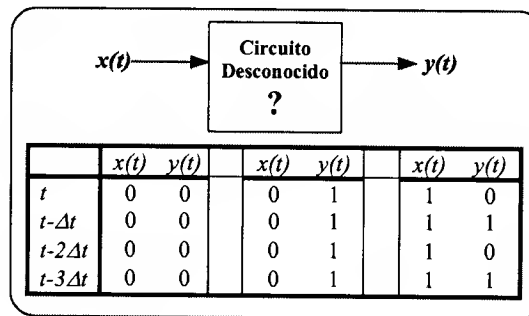
- Espacio de entradas, ( $X = x_1, x_2$ )
- Espacio de estados, ( $S = S_1, S_2$ )
- Espacio de salidas, ( $Y = y_1, y_2, y_3$ )
- Ley de producción de nuevos estados a partir de las entradas y el estado anterior.
- Ley de producción de salidas a partir de las entradas y del estado anterior.

Obsérvese que en esta introducción del concepto de autómatas finitos las variables del espacio de entradas, el espacio de salidas y el conjunto de estados pueden tomar más de dos valores. Más adelante, cuando proyectemos el concepto general de autómatas en su realización física mediante circuitos digitales, tendremos que codificar todas las variables en binario.

El punto importante a nivel conceptual en la teoría de autómatas es comprender que hay necesidad de *tantos estados* internos en un circuito como *historias equivalentes de estímulos*. Cada historia distinguible de entradas exige un nuevo estado interno. Y lo mínimo son dos estados distintos. Por eso para su síntesis necesitamos circuitos capaces de acomodar de forma permanente dos estados eléctricos distintos.

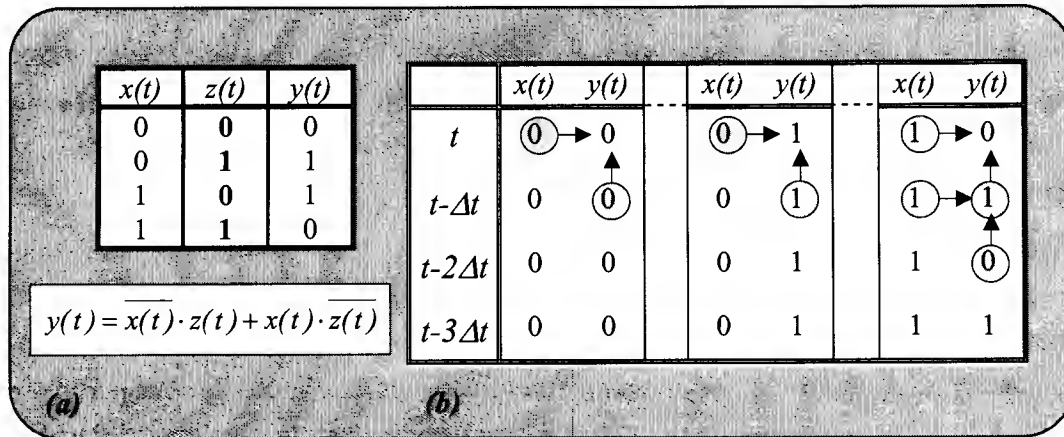
Todo lo que puede pasar en la vida de un bolígrafo es que su punta esté dentro o fuera y por muchas veces que lo use sólo llegará a una de estas dos situaciones finales: punta dentro o punta fuera. Por eso sólo necesito *dos estados* ( $S_1$  y  $S_2$ ) para describir su dinámica. Ahora bien, ¿qué pasa si queremos describir la conducta de 2 bolígrafos?. Necesitaremos entonces un autómata con 4 estados (00, 01, 10, 11) porque pueden existir cuatro configuraciones mutuamente exclusivas (los dos con la punta dentro, uno dentro y otro fuera, al revés y los dos con la punta fuera).

Veamos ahora de otra forma la necesidad de introducir retardos o “memoria” (variables que describen el estado interno de un circuito). Al analizar el circuito de la figura con una sola entrada,  $x(t)$ , y una sola salida,  $y(t)$ , para obtener la función que realiza,  $y(t) = f[x(t)]$ , se encuentran las siguientes correspondencias entre secuencias temporales de entrada y salida que se ilustra en la tabla de la *figura 8.2* y nos piden que encontremos cuál es el circuito combinacional que hay dentro de la “caja negra”.



**Figura 8.2.** Circuito no conocido del que sólo podemos medir las salidas correspondientes a distintas secuencias de entrada

Para empezar, no hay ninguna función combinacional,  $y=f(x)$ , que genere estas tablas, ya que unas veces ante “0” da “0” y otras veces ante “0” da “1”. Parece lógico pensar entonces que la salida  $y(t)$  no sólo depende de  $x(t)$  sino de alguna otra variable interna,  $z(t)$ , que no es observable. Así, podríamos explicar estos resultados con una tabla como la de la *figura 8.3.a*.



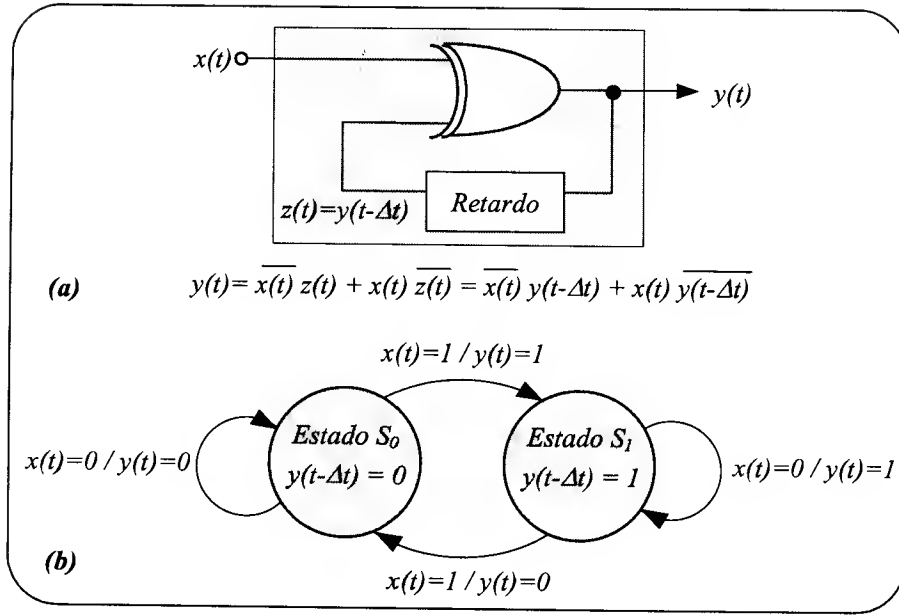
**Figura 8.3.** (a) Posible solución introduciendo una nueva variable  $z(t)$ . (b) Búsqueda de una ley en las secuencias de valores de entrada y salida.

Pero ¿quien es esa “misteriosa” variable interna  $z(t)$ ? Si observamos las secuencias de valores marcadas en las tablas (que repetimos por comodidad) vemos que (*figura 8.3.b*):

1. Si  $x(t) = 0$  e  $y(t-\Delta t) = 0$ , entonces  $y(t) = 0$
2. Si  $x(t) = 0$  e  $y(t-\Delta t) = 1$ , entonces  $y(t) = 1$
3. Si  $x(t) = 1$  e  $y(t-\Delta t) = 0$ , entonces  $y(t) = 1$
4. Si  $x(t) = 1$  e  $y(t-\Delta t) = 1$ , entonces  $y(t) = 0$

Así pues, la misteriosa variable  $z(t)$  es simplemente la salida,  $y(t)$ , retardada  $\Delta t$ , es decir  $z(t) = y(t-\Delta t)$ , con lo que el circuito desconocido junto con su diagrama de transición de estados es el que se ilustra en la *figura 8.4*.

Evidentemente, como ya habrá detectado el lector, esta es la función del biestable T que usamos previamente, en forma de bolígrafo, para introducir el concepto de estado.



**Figura 8.4.** (a) Circuito con una puerta XOR y un retardo. (b) Diagrama de transición de estados.

Hemos usado estos ejemplos tan sencillos porque contienen todos los elementos necesarios para formular la teoría de autómatas. En términos más formales diremos que un autómata finito y determinístico es una máquina matemática que opera en una escala cuantificada de tiempos y viene definido por un quinteto:

$$A = (X, Y, S; f, g)$$

[8.1]

donde:

$X$ : es el conjunto finito de posibles entradas,  $\{x_i\}$ ,  $i = 1, \dots, n$

$Y$ : es el conjunto finito de posibles salidas,  $\{y_k\}$ ,  $k = 1, \dots, p$

$S$ : es el conjunto finito de posibles estados internos,  $\{S_j\}$ ,  $j = 1, \dots, m$

$f$  y  $g$ : son dos conjuntos de reglas de decisión que representan la dinámica del sistema en la producción de nuevos estados (reglas  $f$ ) y en la producción de salidas (reglas  $g$ )

La función de producción de nuevos estados,  $f$ , se define en extenso como una aplicación del producto cartesiano  $X \times S$  sobre  $S$  y la función de producción de salidas,  $g$ , es una aplicación del

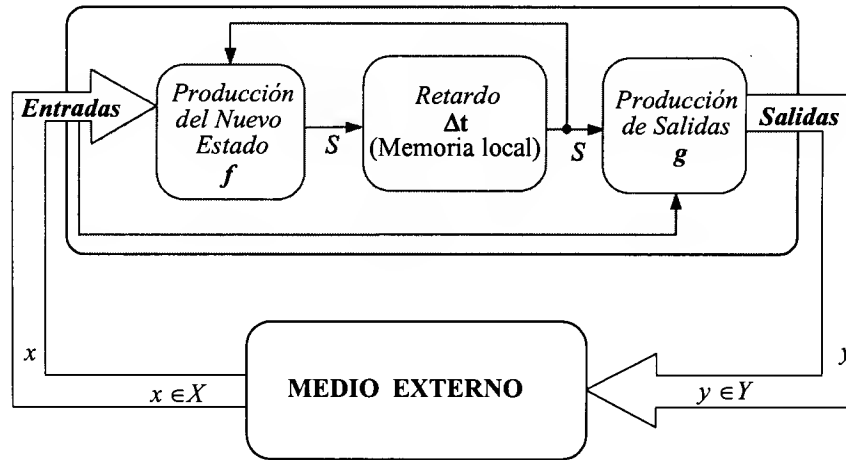
producto cartesiano  $X \times S$  sobre  $Y$ .

$$X \times S \xrightarrow{f} S : S(t + \Delta t) = f[x(t), S(t)] \quad [8.2]$$

$$X \times S \xrightarrow{g} Y : y(t + \Delta t) = g[x(t), S(t)] \quad [8.3]$$

Es decir, el nuevo estado,  $S(t + \Delta t)$ , es una función,  $f(x, S)$ , del estado actual,  $S(t)$ , y de la entrada,  $x(t)$ . Ante una misma entrada, pueden producirse distintas transiciones de estado. A su vez, las mismas entradas  $x(t)$  y el mismo estado  $S(t)$  participan a través de otra función,  $g(x, S)$ , en la producción de las salidas. Estas variables de salida son las respuestas del autómata a un medio externo que lo ha activado a través de las entradas,  $x(t)$ . Es decir, un autómata finito siempre se interpreta en su relación con un medio externo de forma que las salidas del autómata son las entradas del medio y viceversa.

Es interesante comparar el esquema de la *figura 8.5* con el modelo computacional que propusimos como introducción al libro en la *figura 1.1* porque ambos pretenden contestar de forma intuitiva no sólo a las necesidades lógico-matemáticas de la electrónica digital, sino a la cuestión más amplia sobre cuál es el modelo de sistema físico capaz de computar cualquier función computable. Enlazamos así la teoría de autómatas con las máquinas de Turing, y los lenguajes formales que constituyen el soporte teórico-formal de muchos aspectos esenciales dentro del estudio de los lenguajes de programación.

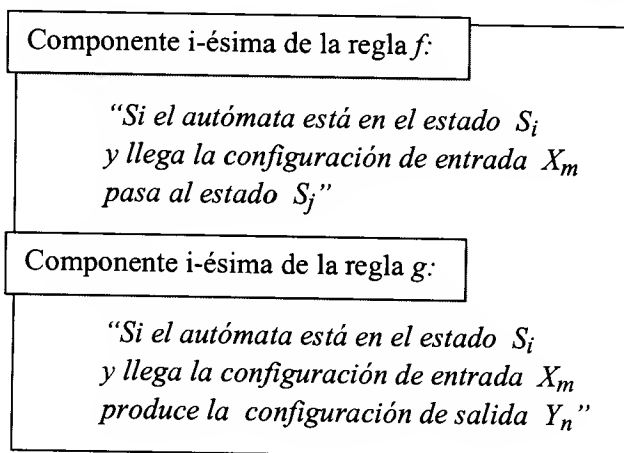


**Figura 8.5** Estructura computacional de un autómata finito y determinístico.

Una vez establecida la conexión con la teoría matemática de la computación vamos a volver al uso limitado que se hace de la misma en el campo de la electrónica digital. En su formulación general, las entradas,  $\{x_i\}$ , las salidas,  $\{y_k\}$ , y los estados,  $\{S_j\}$  son los símbolos, cuyo significado se establece por convenio mediante una tabla de semántica. En electrónica digital, las entradas, los estados y las salidas son niveles lógicos, como el resto de las señales que hemos manejado en los siete primeros temas dedicados al estudio de la lógica combinatorial. En todos los casos, además, el número de configuraciones de valores posibles es del tipo  $2^n$ . Con una entrada,  $x_1$ , tenemos dos valores (0,1), con dos entradas,  $(x_1, x_2)$ , tenemos cuatro configuraciones de entrada posibles, (00, 01, 10, 11), etc..

Lo mismo pasa con las salidas y los estados. Con una variable de salida,  $y_1$ , tenemos dos valores, (0, 1), con dos  $(y_1, y_2)$ , cuatro valores, etc... Finalmente, el estado del autómata queda representado por las salidas de un conjunto de elementos de memoria a los que llamaremos biestables D (de Delay, retardo). Cada biestable D almacena el estado de una variable binaria,  $s_j$ , que sólo puede tomar dos valores, (0, 1). Por consiguiente el autómata mínimo es aquel que posee una única variable de estado ( $s_0$ ) y, por consiguiente, dos estados, ( $S_0 \rightarrow \bar{s}_0, S_1 \rightarrow s_0$ ). El siguiente se obtiene añadiendo un nuevo biestable ( $s_1$ ), lo que da lugar a cuatro estados ( $S_0 \rightarrow \bar{s}_1 \bar{s}_0, S_1 \rightarrow \bar{s}_1 s_0, S_2 \rightarrow s_1 \bar{s}_0, S_3 \rightarrow s_1 s_0$ ).

El mismo tipo de simplificación se obtiene al intentar concretar de forma digital las expresiones de las reglas de decisión,  $f$  y  $g$ , encargadas de la transición entre estados ( $f$ ) y de la producción de salidas ( $g$ ). La forma más general de describir estas reglas es usar expresiones condicionales del tipo:

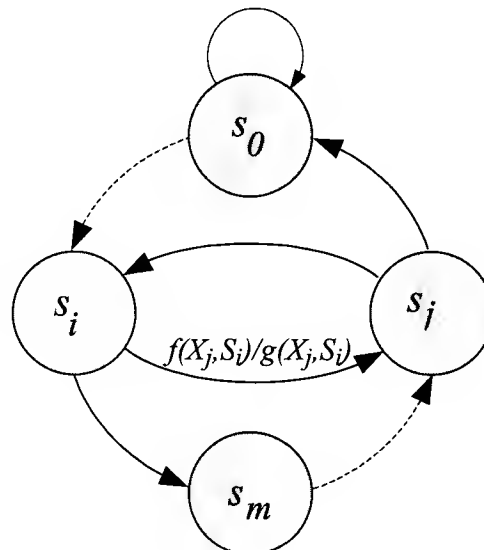




Estas reglas condicionales que describen a  $f$  y a  $g$  se suelen organizar en forma de tablas de transición ( $f$ ) y tablas de producción de estados ( $g$ ) o bien mediante diagramas de transición de estados. Las tablas de transición son matrices simbólicas en las que las columnas corresponden a posibles símbolos de entrada ( $X_1, X_2, \dots, X_n$ ) y las filas a posibles estados ( $S_1, S_2, \dots, S_m$ ). El elemento  $(i, j)$  de esta matriz representa los nuevos estados y nuevas salidas correspondientes,  $f(S_i, X_j)$  y  $g(S_i, X_j)$ . Esta técnica de representación se muestra en la *figura 8.6.a*. Las transiciones de estado correspondientes se ilustran en la *figura 8.6.b*. En este caso los estados están asociados a los *nodos* y las transiciones son los *arcos* que van del estado actual al siguiente, la descripción del arco representa las condiciones en las que se produce esa transición. Obviamente, si no existe ningún símbolo de entrada,  $X_i$ , tal que  $f(X_i, S_j) = S_j$ , no existirá el arco correspondiente en el diagrama de transición de estados diciéndonos que los estados  $S_i$  y  $S_j$  no están conectados bajo ninguna condición.

		Entrada Actual				
Estado Actual		$X_0$	...	$X_j$	...	$X_n$
	$S_0$	$f(X_0,S_0)/g(X_0,S_0)$		...		$f(X_n,S_0)/g(X_n,S_0)$
	$S_i$	...		$f(X_j,S_i)/g(X_j,S_i)$		...
	$S_m$	$f(X_0,S_m)/g(X_0,S_m)$		...		$f(X_n,S_m)/g(X_n,S_m)$

(a)



(b)

**Figura 8.6.** (a) Tabla de transición. (b) Diagrama de transición de estados.

Recordemos que los elementos de los espacios de entrada y salida son en general simbólicos. Cada entrada, estado o salida se asocia a una etiqueta cuya especificación en las tablas de significados expresa en lenguaje natural una descripción de la situación que estamos representando. Así, cada computación general, se descompone en dos procesos: (a) *identificación de los conjuntos*  $X$ ,  $Y$  y  $S$  necesarios para hacer computacional una determinada tarea y (b) *búsqueda de las funciones*  $f$  y  $g$  ó sus representaciones equivalentes en términos de tablas ó diagramas de transición.

En electrónica digital las funciones  $f$  y  $g$  son funciones en lógica combinacional del tipo estudiado en los temas anteriores de forma que las tablas de transición entre estados y las tablas de producción de salidas son ahora matrices booleanas.

Esta simplificación, obtenida al especificar las entradas, estados y salidas de un autómata finito en electrónica digital, nos permite volver a dibujar la estructura de la *figura 8.5* de otra forma equivalente más próxima a su síntesis electrónica. El resultado se muestra en la *figura 8.7* donde hemos cambiado el nombre de autómata finito por el de circuito secuencial. Un circuito secuencial de  $M$  variables de entrada,  $(x_0, x_1, \dots, x_{M-1})$ ,  $N$  variables de salida  $(y_0, y_1, \dots, y_{N-1})$  y  $Q$  estados (siendo  $Q \geq 2^N$ ), se puede sintetizar de forma modular de acuerdo con el esquema de la *figura 8.7*.

Hay una función múltiple en lógica combinacional que sintetiza las  $N$  variables de estado,  $(D_0, D_1, \dots, D_{N-1})$  a partir de los valores de las  $M$  variables de entrada,  $(x_0, x_1, \dots, x_{M-1})$  y de los valores del estado en el intervalo anterior,  $(Q_0, Q_1, \dots, Q_{N-1})$ .

$$D_i(t) = f_i[x_0(t), \dots, x_{M-1}(t); Q_0(t), \dots, Q_{N-1}(t)] \quad [8.4]$$

Siendo,

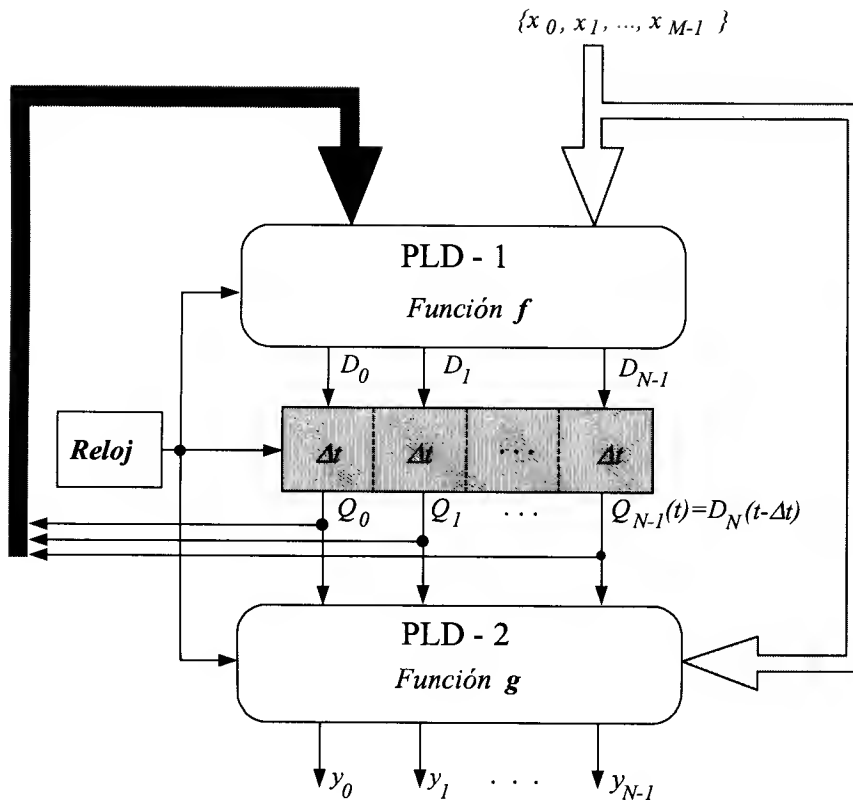
$$Q_i(t) = D_i(t - \Delta t), \quad i = 0, 1, \dots, N - 1 \quad [8.5]$$

Análogamente, las  $N$  variables de salida,  $(y_0, y_1, \dots, y_{N-1})$  son en general otra función múltiple de las entradas  $(x_0, x_1, \dots, x_{M-1})$ , y del estado  $(Q_0, Q_1, \dots, Q_{N-1})$ .

$$y_j(t) = g_j[x_0(t), \dots, x_{M-1}(t); Q_0(t), \dots, Q_{N-1}(t)]$$

$$j = 0, 1, \dots, N - 1 \quad [8.6]$$

La síntesis de las funciones combinacionales  $\{f_i\}$  y  $\{g_i\}$  puede realizarse usando PLDs que incluyen en sus macroceldas biestables para la síntesis de los retardos.



**Figura 8.7.** Arquitectura general de un circuito secuencial. Es decir, de un autómata finito en el que las variables son pulsos o niveles, las funciones son lógica combinacional y el estado se almacena en  $N$  biestables.

Para muchas aplicaciones no es necesario usar las funciones  $\{g_i\}$ , de forma que las variables de salida,  $\{y_i\}$ , coinciden con las variables de estado,  $\{Q_i\}$ . En estas situaciones se puede realizar de forma sencilla la *síntesis modular* de cualquier circuito secuencial de  $2^N$  estados. Cada una de sus  $N$  variables de estado,  $(y_0, y_1, \dots, y_{N-1})$ , consta de una función combinacional,  $D_i = f_i$ , y un retardo,  $y_i(t) = D_i(t + \Delta t)$ . Esta combinación puede considerarse como un *circuito secuencial mínimo y universal* de  $(M+N)$  entradas ya que posee una *función combinacional universal* de  $(M+N)$  entradas y dos estados. Veremos en el tema final del libro, al introducir la lógica secuencial programable, cómo se basa en esta idea de *síntesis modular* usando circuitos universales. Aquí sólo nos interesa introducir la idea al plantear el modelo matemático de la electrónica secuencial.

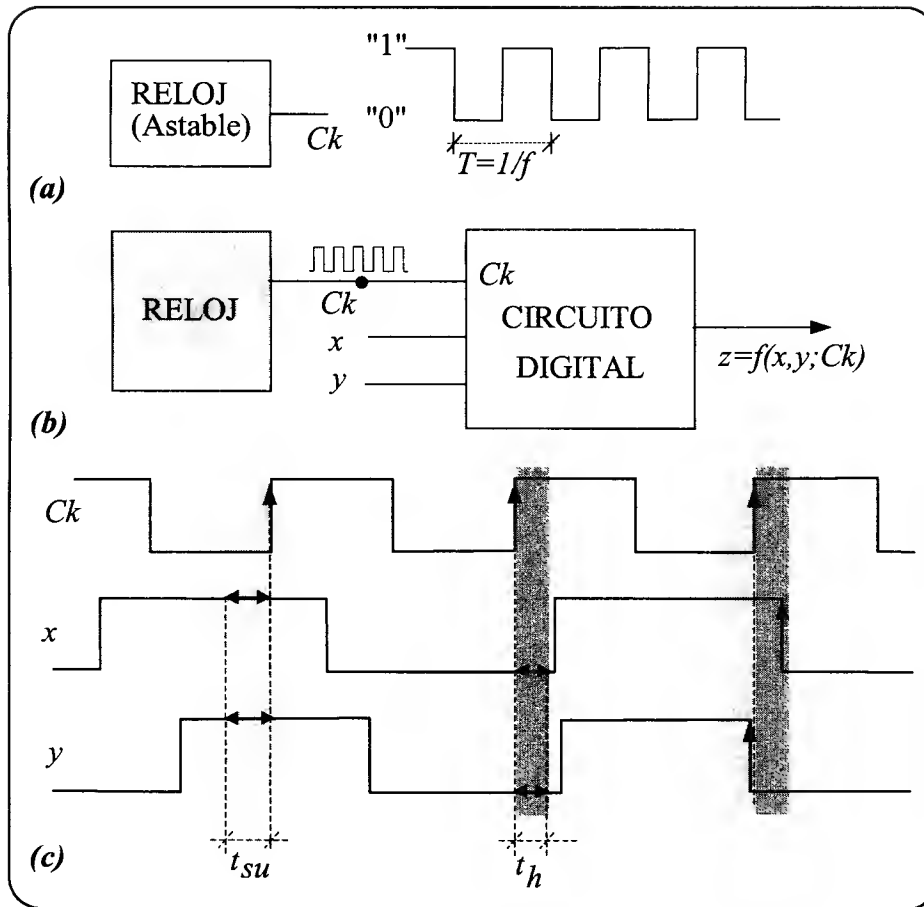
## 8.2. EL TIEMPO EN DIGITAL: COMPORTAMIENTO SÍNCRONO Y ASÍNCRONO

En electrónica combinacional no existía el tiempo, la computación era "instantánea". En cambio, en electrónica secuencial el tiempo es esencial. La posición relativa en la que ocurren los sucesos define la computación. Hay dos tipos de circuitos digitales, *síncronos* y *asíncronos*.

Son *circuitos síncronos* aquellos que están gobernados por un reloj central de forma que todos los sucesos de interés computacional ocurren en los entornos de los pulsos de reloj. El intervalo entre dos pulsos sucesivos se usa para estabilizar los niveles de tensión, pero los cambios de estado no se producen hasta que no llega el nuevo pulso. Más específicamente, la conmutación se produce en los flancos del pulso de reloj, cuando este pasa de baja a alta (flanco positivo) o viceversa (flanco negativo). La *figura 8.8* ilustra el funcionamiento y los problemas asociados a un sistema síncrono. Todos los circuitos secuenciales síncronos poseen junto a las entradas necesarias para realizar su función, una entrada adicional (*Ck* - clock) para la señal de reloj que es la que controla los instantes en que se hacen efectivos los cambios que definen la función.

Un reloj es un oscilador, un circuito astable que genera un tren de pulsos o una onda cuadrada. En el tema correspondiente a temporizadores estudiaremos su diseño y las soluciones integradas monofásicas y polifásicas. Aquí nos basta con la idea intuitiva de oscilador. El intervalo temporal entre dos pulsos sucesivos es el periodo ( $T$ ) y su inverso la frecuencia ( $f=1/T$ ). La frecuencia del reloj define la máxima velocidad de operación permitida en un sistema digital. Cualquier función necesita, al menos, un periodo (dos pulsos sucesivos) para ejecutarse.

La parte (c) de la *figura 8.8* ilustra las restricciones en las señales de entrada para conseguir un buen funcionamiento síncrono. Antes de cada subida del pulso de reloj, las señales  $x$  e  $y$  deben tener valores estables. Hay un intervalo de seguridad llamado *tiempo de asentamiento* (setup time,  $t_{su}$ ) que garantiza que las señales de entrada ya han alcanzado su estado estacionario un tiempo  $t_{su}$  antes de la subida del pulso. De forma complementaria, se define el *tiempo de retención* (hold time,  $t_h$ ) como el intervalo de tiempo durante el cual las entradas todavía tienen que permanecer estables después de haberse producido la subida de baja a alta del pulso de reloj. Si no se cumplen las restricciones de estos tiempos, ( $t_{su}$  y  $t_h$ ), puede producirse un comportamiento errático, no predecible.

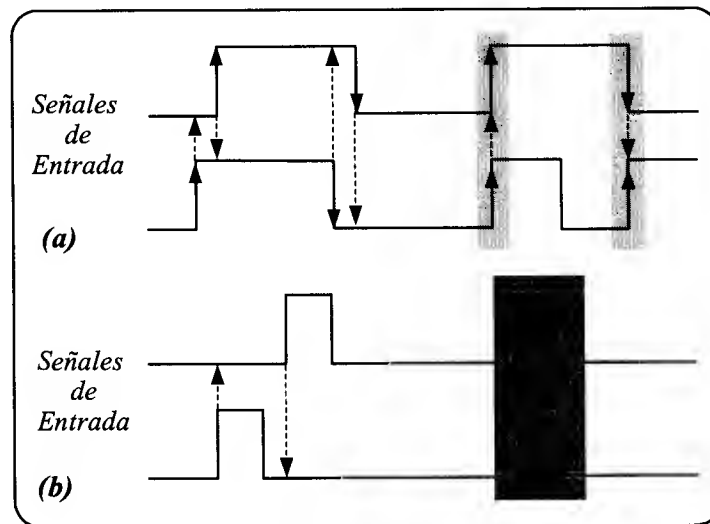


**Figura 8.8.** Comportamiento síncrono. (a) Reloj. (b) Circuito digital con la entrada adicional  $Ck$ . (c) Cronograma ilustrando los tiempos de asentamiento ("setup time") y de retención ("hold time").

El resto del tiempo, dentro de un periodo, desde  $(T_n + t_h)$  hasta  $(T_{n+1} - t_{su})$ , es el intervalo que queda para la transición de estado en las señales externas. Obviamente cuanto mayor sea la frecuencia del reloj ( $f$ ), menor será el periodo ( $T=1/f$ ) y menor el tiempo de transición. Por consiguiente, más rápida tiene que ser la familia lógica usada para la síntesis de las funciones combinacionales que cambian el valor de las entradas.

La mayoría de los sistemas digitales son síncronos y así serán todos los estudiados en este curso. Sin embargo, existen aplicaciones que optimizan el tiempo usando circuitos asíncronos, en los

que no se depende de un reloj de forma que a medida que se producen los cambios en las variables de entrada, repercuten en la salida, sin esperar la llegada de un nuevo pulso de reloj. Los sistemas asíncronos son más difíciles de tratar que los síncronos, tanto a nivel matemático como a nivel de realización física. La *figura 8.9* ilustra algunas de las restricciones del diseño asíncrono. Hay dos modos posibles de operación: *fundamental* y *por pulsos*. En el *modo fundamental* no pueden existir cambios simultáneos de nivel en las variables externas. Sólo una puede estar conmutando en cada instante de tiempo. En el *modo de pulsos* las señales sólo están en alta durante un corto intervalo de tiempo. La subida del pulso marca el suceso temporal y, al igual que antes, no pueden coincidir dos pulsos a la vez.



**Figura 8.9.** Circuitos asíncronos. *(a)* Modo fundamental. *(b)* Funcionamiento por pulsos.

### 8.3. BIESTABLES

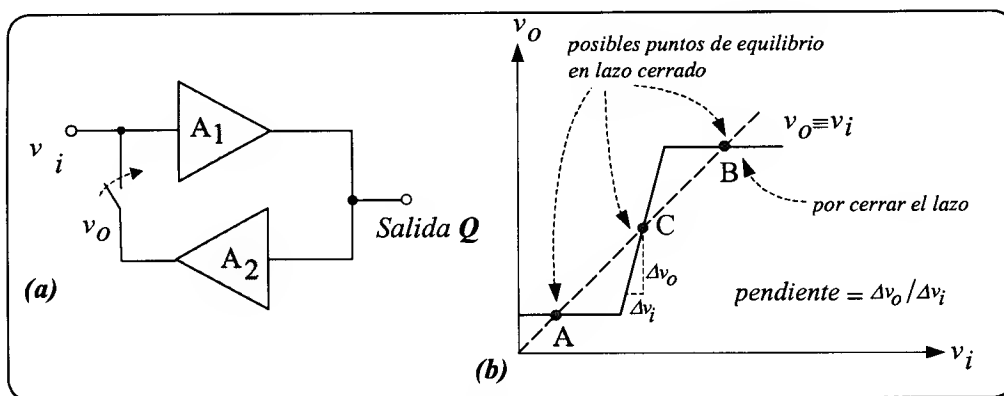
Si observamos las *figuras 8.5* y *8.7* que resumen el modelo matemático de la electrónica secuencial vemos que para su síntesis tenemos todo lo que necesitamos, salvo los retardos, a los que hemos llamado *biestables* D. Estos circuitos biestables son los encargados de almacenar el estado interno del sistema y de ellos vamos a hablar ahora viendo cuál es su fundamento electrónico y cómo se sintetizan usando puertas lógicas NAND ó NOR.

Los circuitos biestables son circuitos binarios (con dos estados) en los que ambos estados son estables de forma que hace falta una señal externa de excitación para hacerlos cambiar de estado. Esta función de excitación define el tipo de biestable (D, T, R-S ó J-K) y de ella empezaremos a hablar en el apartado siguiente.

Desde una perspectiva electrónica, son circuitos *binarios* aquellos que poseen dos estados internos distinguibles, en general asociados al estado de corte o saturación de los transistores bipolares o MOS con los que se han sintetizado los inversores. Los circuitos binarios pueden ser *biestables*, *monoestables* o *astables* (multivibradores) dependiendo de que ambos estados sean estables (biestables), lo sea uno sólo (monoestable) o no lo sea ninguno de los dos y el circuito esté cambiando constantemente de estado (astable). En este tema nos interesa estudiar los circuitos biestables porque son la base de los elementos de memoria. Más adelante estudiaremos los monoestables y los astables que son la base de los circuitos temporizadores y de los relojes.

Para la síntesis de circuitos binarios biestables usamos dos amplificadores inversores de ganancia  $A_1$  y  $A_2$  conectados de forma tal que cierran un lazo de *realimentación positiva*, como se ilustra en la *figura 8.10.a*. Es decir, la salida del primer amplificador es la entrada del segundo y viceversa. La salida,  $v_o$ , antes de cerrar el lazo será

$$v_o = A_1 \cdot A_2 \cdot v_i = A_v \cdot v_i \quad [8.7]$$



**Figura 8.10.** Circuitos biestables. (a) Realimentación positiva de dos inversores. (b) Característica de transferencia.

Si la ganancia total  $A_v = A_1 \cdot A_2$  es mayor que la unidad hay una ganancia neta en todo el

recorrido de la señal al cerrar el lazo. La señal que vuelve es mayor o igual que la aplicada, de forma que el circuito entra en una zona de alta ganancia pero inestable. Aquí permanece sólo transitoriamente empezando una evolución hacia uno de los dos estados estables posibles, tal como se ilustra en la *figura 8.10.b*:

- O bien decrece la ganancia en uno de los amplificadores inversores porque entra en saturación
- O se abre el lazo de transmisión porque el elemento activo (transistor) de uno de los inversores pasa al corte.

La parte (b) de la *figura 8.10* ilustra los tres únicos puntos que podrían considerarse en principio puntos de equilibrio (A, B y C). Pero sólo A y B son estables. En efecto, en C la ganancia en tensión de los inversores ( $A_v = \Delta v_o / \Delta v_i$ ) es mucho mayor que la unidad y cualquier desequilibrio o señal de ruido es amplificado llevando al circuito al punto B (inversor de salida en corte y ganancia nula) o al punto A (inversor de salida saturado), y en ambos casos la ganancia incremental es nula y el estado final es estable.

Así pues, las condiciones necesarias y suficientes para el funcionamiento biestable de dos inversores realimentados son:

- Existencia de *dos estados estables* en continua.
- La *ganancia en lazo cerrado* ( $A_v$ ), con ambos inversores en zona activa, debe ser mayor que la *unidad* en un punto de funcionamiento posible en continua (C), siendo

$$A_v = A_1 \cdot A_2 = \frac{\Delta v_{o1}}{\Delta v_{i1}} \cdot \frac{\Delta v_{o2}}{\Delta v_{i2}}, \quad v_{o1} = v_{i2}, \quad v_{i1} = v_{o2} \quad [8.8]$$

El paso de un estado estable al otro se realiza con aporte de energía exterior (pulso o cambio de nivel) que permite superar el punto en el que la ganancia del lazo es  $A_v = 1$ .

Hay cuatro tipos básicos de biestables en función de cual es la configuración de las variables externas que controlan las transiciones de estado: R-S, J-K, T ó D. Además, dentro de cada uno de estos tipos de comportamiento externo, los biestables se clasifican en tres grupos:

1. “*Latches*” básicos o con puertas previas con una entrada adicional de pulsos de reloj para facilitar el control de las entradas (*biestables sincronizados a niveles*).
2. *Maestro-esclavo* disparados por pulsos y con separación de la orden de disparo de su ejecución.



### 3. Biestables *disparados por flancos*.

Obsérvese que en cada uno de estos grupos podemos tener configuraciones R-S, J-K, T ó D.

Empezaremos estudiando el núcleo del *biestable R-S básico* y después iremos complicando el circuito con la introducción de puertas, las posibilidades de control síncrono y asíncrono del estado, etc...

Ya hemos visto que la componente estructural básica de todos los biestables (su núcleo) la forman *dos inversores acoplados en realimentación positiva* de forma tal que cuando la salida de uno está en alta ( $Q$ ) las del otro está en baja ( $\bar{Q}$ ). Necesitamos ahora añadirle entradas externas para controlarlo. Es decir, para poder escribir un “1” ó un “0” y, por consiguiente controlar las cuatro transiciones posibles del autómata equivalente de dos estados (pasar de “0” a “1”, de “1” a “0”, dejarlo en “0” ó dejarlo en “1”).

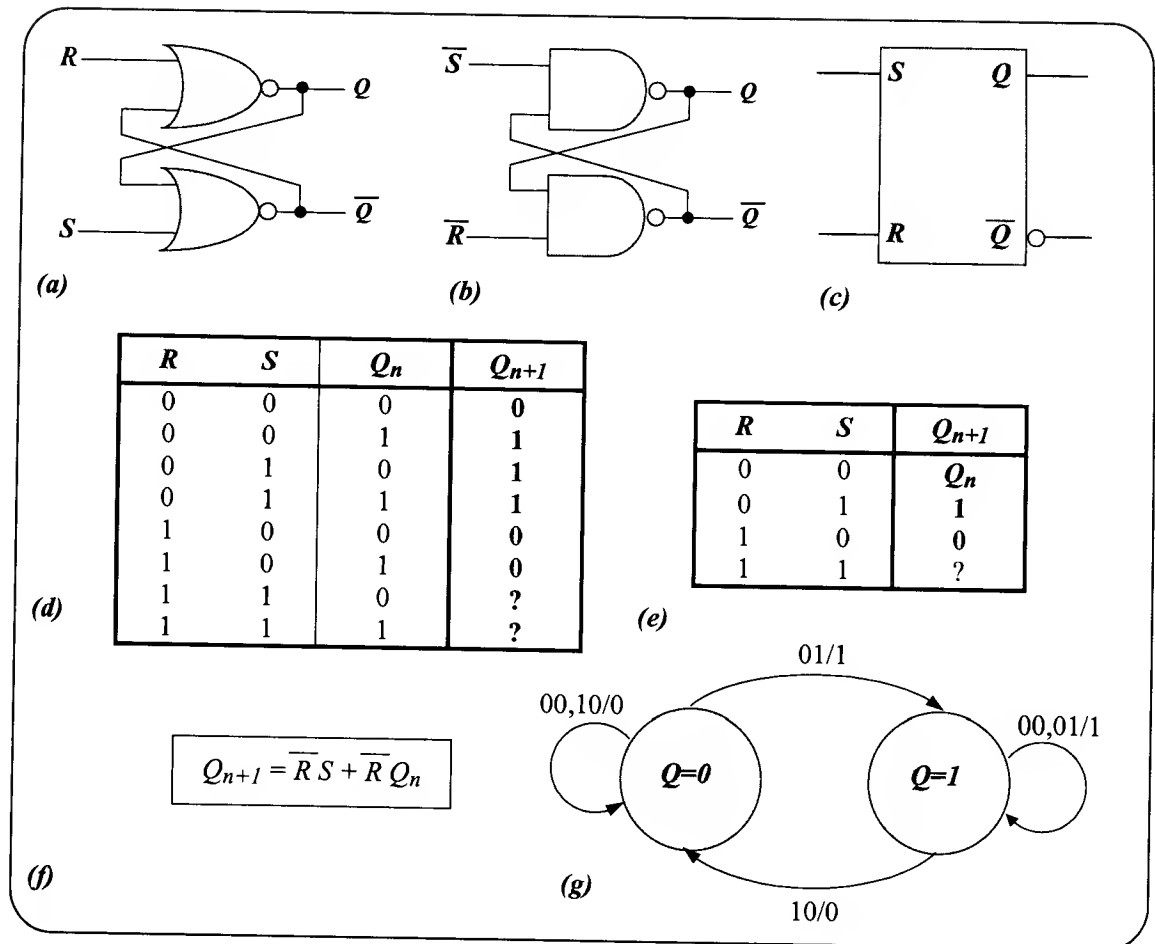
#### 8.3.1. R-S Básico

La configuración que especifica la forma de controlar estas transiciones da nombre al biestable (R-S, J-K, T ó D). Así, la configuración R-S es aquella que tiene dos entradas externas,  $R$  (*Reset* o puesta a “0” del biestable) y  $S$  (*Set* o puesta a “1”) y la expresión más sencilla de esta configuración se consigue realimentando dos puertas NAND o dos puertas NOR de dos entradas, tal como se ilustra en el circuito de la *figura 8.11*. En el caso de las puertas NOR, por convenio, llamamos  $Q$  a la salida de la puerta de arriba, cuya entrada externa es  $R$ , y  $\bar{Q}$  a la salida de la puerta de abajo, cuya entrada externa es  $S$ . Así, debido a la realimentación, en la primera puerta NOR entran  $R$  y  $\bar{Q}$  y en la segunda  $Q$  y  $S$ <sup>(\*)</sup>.

Vamos ahora a analizar en extenso las distintas posibilidades lógicas en las que pueden permanecer estas puertas de forma estable. El análisis será algo más minucioso de lo usual porque es el primero en el que interviene la realimentación y queremos usarlo para ilustrar el método que se repetirá después en otras versiones más complejas de esta configuración y en otras configuraciones.

---

<sup>(\*)</sup> Por claridad en la exposición haremos todo el desarrollo con puertas NOR. En el texto de problemas aparece el desarrollo equivalente usando sólo puertas NAND.



**Figura 8.11.** Síntesis de la versión mínima de los biestables R-S. (a) Con dos puertas NOR realimentadas (entradas R y S activas en alta). (b) Con dos puertas NAND (entradas  $\overline{R}$  y  $\overline{S}$  activas en baja). (c) Símbolo lógico, (d) y (e) Tablas. (f) Expresión lógica. (g) Diagrama de transición de estados.

Empecemos recordando que una puerta NOR sólo está en alta cuando sus dos entradas están en baja ( $00 \Rightarrow 1$ ;  $01, 10, 11 \Rightarrow 0$ ). Por consiguiente, en cuanto una de las entradas esté en alta, la salida es “0”, independientemente del nivel lógico de la otra entrada. Recordemos también que lo que estamos buscando es una expresión lógica que nos relacione el valor de la salida del biestable en un cierto intervalo de tiempo ( $Q(t+\Delta t)$ ) que por simplificación en la escritura llamaremos  $Q_{n+1}$ ) en función de los valores de las entradas de control ( $R_n$  y  $S_n$ ) y del estado (salida en el instante actual, o sea,  $Q_n$ ).

Además, vamos a obtener esta función construyendo la tabla de verdad. Es decir, viendo cuál es el valor estable cuando establecemos como condiciones iniciales los ocho valores posibles de  $R_n$ ,  $S_n$  y  $Q_n$  (000, 001, ..., 111). La *figura 8.12* ilustra el proceso de forma detallada.

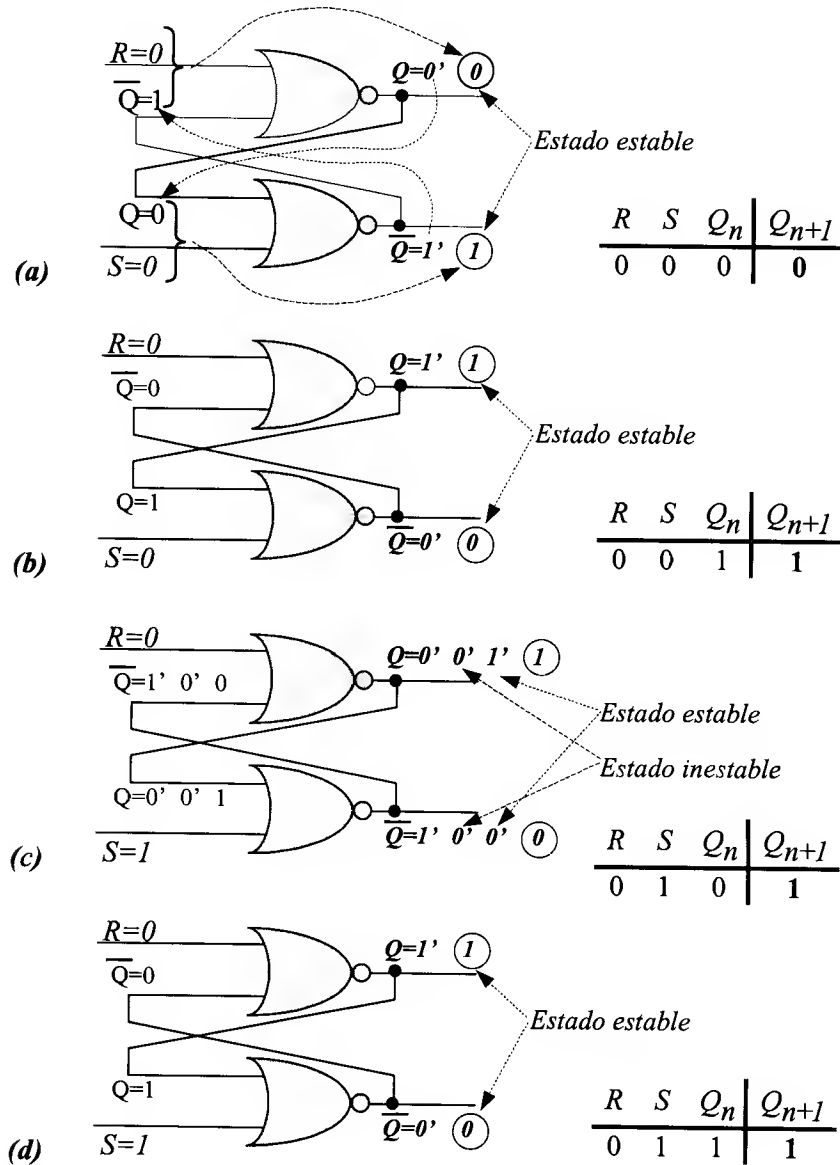
1.  $R_n = 0, S_n = 0$ : En este caso,  $Q_n$  podría estar inicialmente en “0” ó en “1”. Supongamos que  $Q_n = 0$  (*figura 8.12.a*). Entonces, por complementariedad forzosa,  $\bar{Q}_n = 1$  (*figura 8.12.a*). Así, en la primera NOR aparece  $R_n = 0$  y  $\bar{Q}_n = 1$  y, por consiguiente,  $Q_{n+1} = 0$ . Al mismo tiempo, en la segunda NOR aparece  $Q_n = 0$  y  $S_n = 0$ , por consiguiente  $\bar{Q}_{n+1} = 1$ . Es decir, se confirma el estado inicial supuesto, las salidas  $Q_n$  y  $\bar{Q}_n$  son estables y ya tenemos la primera línea de la tabla ( $R_n = 0, S_n = 0, Q_n = 0 \Rightarrow Q_{n+1} = 0$ ).

Supongamos ahora que el estado inicial era  $Q_n = 1$  y, por complementariedad,  $\bar{Q}_n = 0$  (*figura 8.12.b*). Al realimentar, la primera NOR tiene 00 y la segunda 10, con lo que se confirma el estado inicial y tenemos la segunda línea de la tabla ( $R_n = 0, S_n = 0, Q_n = 1 \Rightarrow Q_{n+1} = 1$ ). Es decir, cuando  $R_n = S_n = 0$ , el *biestable no cambia de estado*. Mantiene el estado anterior ( $Q_{n+1} = Q_n$ ).

2.  $R_n = 0, S_n = 1$ : Empecemos también suponiendo que  $Q_n = 0$  y  $\bar{Q}_n = 1$  (*figura 8.12.c*). Así, en la primera puerta NOR aparece  $R_n = 0$  y  $\bar{Q}_n = 1$  con lo que su salida,  $Q_{n+1} = 0$ . Al mismo tiempo en la segunda puerta NOR aparece  $Q_n = 0$  y  $S_n = 1$ , con lo que su salida es “0” (en contra de la suposición inicial). Seguimos entonces el lazo de realimentación y presentamos este valor transitorio de  $\bar{Q}_n = 0$  en la primera NOR que ahora tiene en las entradas (00) y produce  $Q_n = 1$ . Al mismo tiempo en la segunda puerta se ha presentado (01) y se vuelve a repetir  $\bar{Q}_n = 0$ . Finalmente, al realimentar este valor “0” de  $\bar{Q}_n$  a la primera puerta NOR, ésta, ante 00, da  $Q_{n+1} = 1$ , que ya es un valor estable. Así, finalmente, ante  $R_n = 0$  y  $S_n = 1$ , si el estado inicial era  $Q_n = 0$ , el nuevo estado estable es  $Q_{n+1} = 1$ , ( $0, 1, 0 \Rightarrow 1$ ).

Si hubiéramos partido de  $Q_n = 1$  (*figura 8.12.d*), es fácil comprobar que la situación es estable. Es decir, ante  $R_n = 0$ , y  $S_n = 1$ , si  $Q_n = 1$ , entonces  $Q_{n+1} = 1$ . Ya tenemos pues dos líneas más de la tabla de verdad y otra parte de la función. Si  $R_n = 0$  y  $S_n = 1$ , el *biestable* pasa a  $Q_{n+1} = 1$ , sea cual

fuere su estado inicial ( $Q_n = 0$  ó  $Q_n = 1$ ). Esta es la *configuración de set* (puerta a "1").



**Figura 8.12.** Descripción detallada del comportamiento del biestable R-S. Ver la explicación de las configuraciones en el texto. Cuando en una entrada o en una salida hay una sucesión de valores ( $0 \rightarrow 0 \rightarrow 1$ ,  $1 \rightarrow 0 \rightarrow 0$ ) significa que la configuración propuesta inicialmente por  $Q$  y  $\bar{Q}$  no será estable. Cuando se alcanza un valor estable se encierra en un círculo.

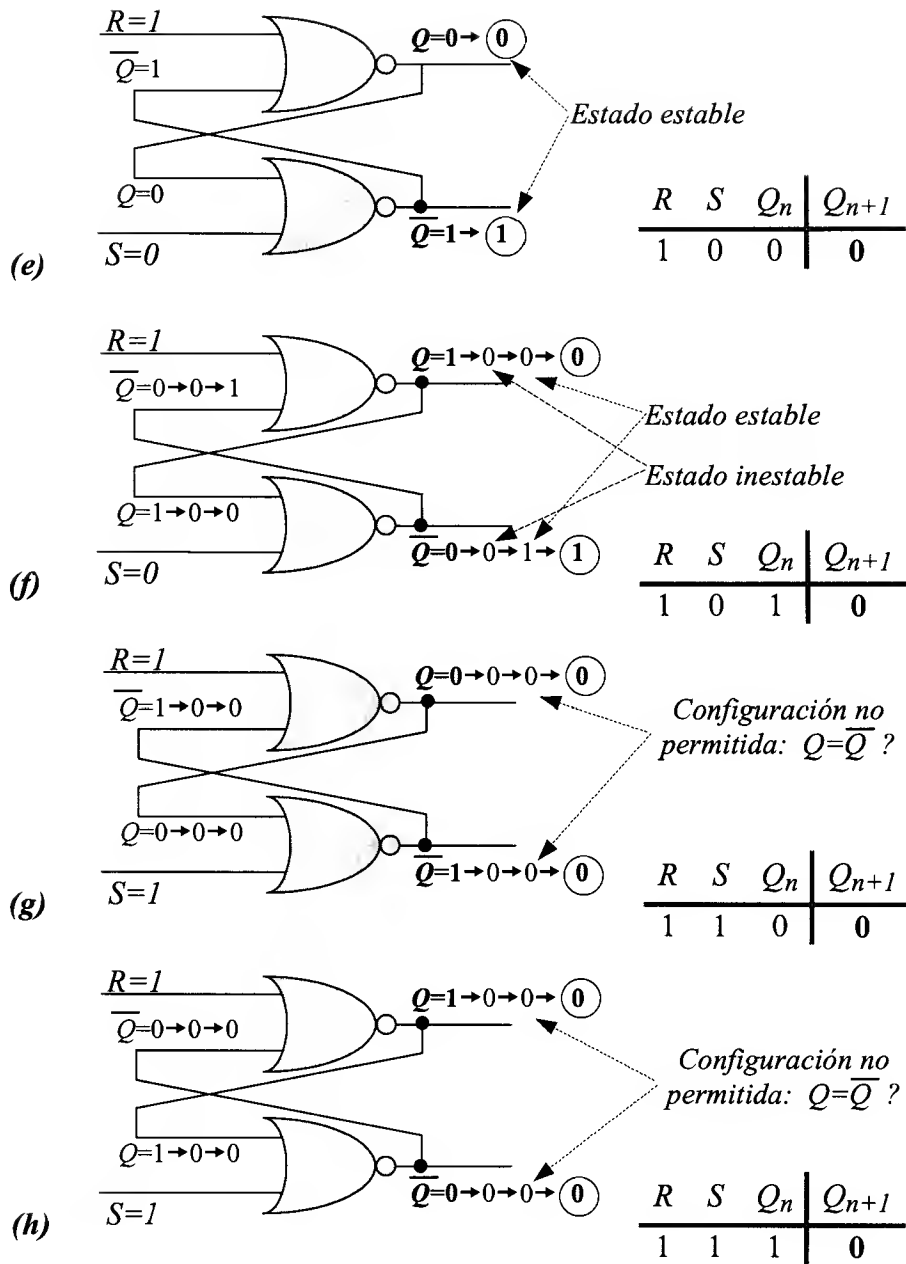


Figura 8.12. (continuación)

3.  $R_n = 1, S_n = 0$ : Supongamos que inicialmente  $Q_n = 0$  (figura 8.12.e).. Es fácil comprobar que en este caso la configuración de salida es estable y  $Q_{n+1} = 0$ , con  $\bar{Q}_{n+1} = 1$ . ( $1\ 0\ 0 \Rightarrow 0$ )

En cambio, si el estado inicial hubiera sido  $Q_n = 1$  (figura 8.12.f), realimentándolo y viendo la sucesión de transitorios que genera veríamos que el único estado final estable es  $Q_{n+1} = 0$  ( $101 \Rightarrow 0$ ). con lo que la ley ahora es: Si  $R_n = 1$  y  $S_n = 0$ , sea cual fuere el estado anterior ( $Q_n = 0$  ó  $Q_n = 1$ ) el nuevo estado es  $Q_{n+1} = 0$ . Esta es la *configuración de reset* (puesta a “0”).

En cambio, si el estado inicial hubiera sido  $Q_n = 1$  (figura 8.12.f), realimentándolo y viendo la sucesión de transitorios que genera veríamos que el único estado final estable es  $Q_{n+1} = 0$  ( $101 \Rightarrow 0$ ). con lo que la ley ahora es: Si  $R_n = 1$  y  $S_n = 0$ , sea cual fuere el estado anterior ( $Q_n = 0$  ó  $Q_n = 1$ ) el nuevo estado es  $Q_{n+1} = 0$ . Esta es la *configuración de reset* (puesta a “0”).

4.  $R_n = 1, S_n = 1$ : Finalmente, si  $R_n = S_n = 1$ , vamos a ver que hay conflicto, por lo que esta *configuración no está permitida* en los biestables R-S.

Supongamos inicialmente que  $Q_n = 0$ , en este caso al realimentar se genera la secuencia que ilustramos en la parte (g) de la figura 8.12 que termina con la configuración “estable”  $Q_{n+1} = \bar{Q}_{n+1} = 0$ . Y lo mismo ocurre en el caso complementario (parte (h) de la figura 8.12) en el que hemos supuesto que el estado inicial era  $Q_n = 1$ . Al realimentar y seguir la secuencia de entradas encontramos que la única configuración estable es  $Q_{n+1} = \bar{Q}_{n+1} = 0$ , lo cual no es posible por la complementariedad forzosa de todo estado biestable.

Así, la expresión de  $Q_{n+1}$  en función de las entradas ( $R_n, S_n$ ) y del estado  $Q_n$ , obtenida a partir de la tabla de la figura 8.11 prohibiendo la entrada (11), es:

$$Q_{n+1} = (S_n + Q_n) \cdot \bar{R}_n \quad [8.10]$$

Con la prohibición de  $R_n = S_n = 1$

**Ejercicio:** Usar un simulador tipo PSpice para comprobar el comportamiento descrito anteriormente de la configuración R-S.

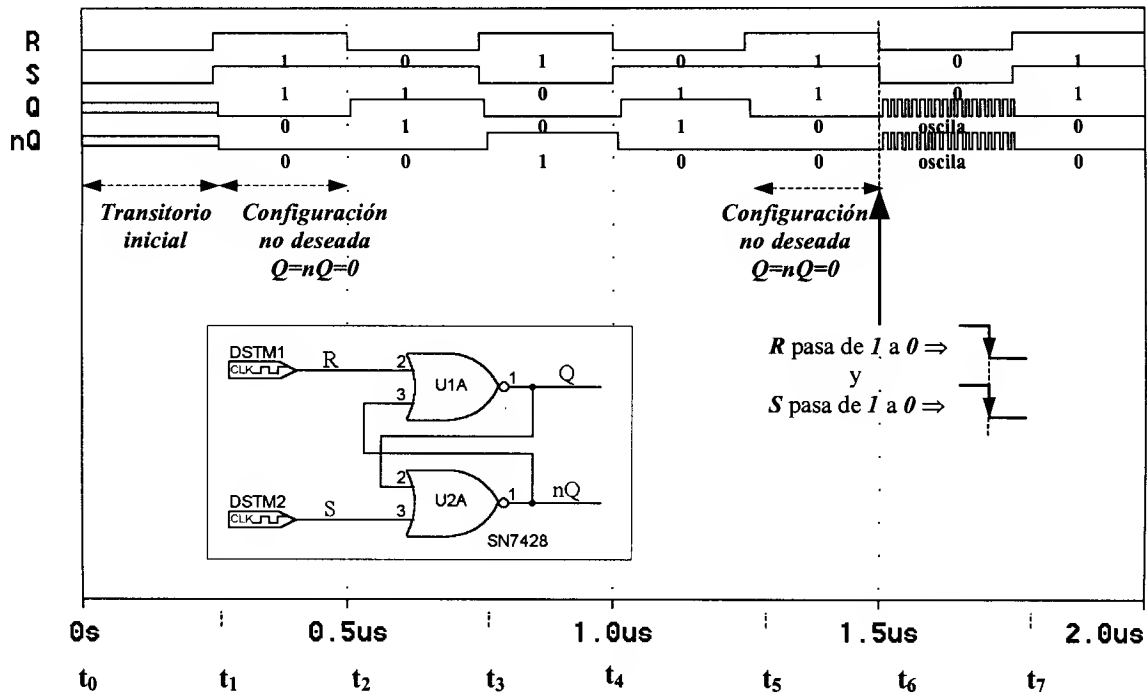
**Solución:**

Usamos puertas tipo SN7428 y dos relojes de distinta frecuencia para obtener de forma

sucesiva las cuatro configuraciones lógicas (00,01,10,11) para las entradas  $R$  y  $S$  y obtenemos los resultados de la *figura 8.13* donde se muestra el circuito simulado y el cronograma.

Tras un transitorio inicial (de  $t_0$  a  $t_1$ ) las dos entradas pasan a 1 (en  $t_1$ ). Con esta configuración de entrada ( $R=S=1$ ) estamos en la situación conflicto en la que tanto la salida  $Q$  como la  $\bar{Q}_n$  (que representamos por  $nQ$ ) están a cero. Esta es la configuración de entrada que nos crea *indeterminación* y que por tanto está prohibida.

En  $t_2$  la entrada  $R$  pasa a 0 mientras que la  $S$  se queda en 1, dando lugar a las salidas  $Q=1$  y  $nQ=0$ , como era de esperar ya que está activa la señal de “set”, es decir,  $S=1$ . En el siguiente intervalo de tiempo ( $t_3$ ),  $R$  pasa a 1 y  $S$  pasa a 0, lo que hace que el biestable cambie de estado poniéndose a 0, es decir,  $Q=0$  y  $nQ=1$ , puesto que está activa la señal de “reset”, es decir,  $R=1$ .



**Figura 8.13.** Simulación del circuito R-S básico. La oscilación aparece cuando  $R$  y  $S$  pasan a la vez de 11 a 00. Si se desea evitar, basta cambiar los relojes (distinto periodo para  $S$  que para  $R$ ) para que no ocurra esa situación.

De nuevo en  $t_5$  vuelve a presentarse la configuración de entrada  $R=1$  y  $S=1$ , produciendo a la salida la configuración no deseada  $Q=0$  y  $nQ=0$ . En el siguiente intervalo de tiempo,  $t_6$  observamos que las dos señales *bajan a la vez* pasando de la configuración de entrada  $11$  a la  $00$  y el circuito empieza a oscilar. Esta oscilación permanece hasta que hay un cambio en las entradas. Su origen está en que las dos puertas NOR “intentan” mantener un estado distinto. Esta oscilación terminaría en el caso de que las dos puertas no fueran exactamente iguales y entonces una de ellas “ganaría” y llevaría al circuito a un estado estable.

Generalmente esto no ocurre en el laboratorio real donde las puertas no son exactamente iguales y siempre hay alguna pequeña diferencia, bien a nivel del circuito o bien a nivel de las señales de entrada (ruido), de forma que es muy poco probable que las dos entradas cambien de  $1$  a  $0$  en el mismo instante de tiempo. Así, en el momento que hay un desfase en los cambios de las señales, una de las puertas cambia de estado antes que la otra y fuerza al circuito a conseguir un estado estable. Sin embargo, en la simulación estamos usando los modelos matemáticos de los circuitos y por tanto las puertas ahora sí que son exactamente iguales y además no hay ningún ruido que interfiera con las señales de entrada.

La forma más sencilla de resolver este problema a nivel de simulación es evitar que las dos señales pasen a la vez de  $1$  a  $0$ . Esto se puede comprobar de forma sencilla sin más que variar la frecuencia de uno de los relojes que se han usado para generar de forma automática las configuraciones de entrada.

---

### 8.3.2. R-S Sincronizado a Niveles

En la configuración R-S básica las transiciones de estado pueden ocurrir en cualquier momento, en función del valor que tomen en ese momento los niveles de tensión en las entradas de set ( $S$ ) y reset ( $R$ ). En este sentido el circuito es asíncrono.

Para acotar con más precisión los intervalos de tiempo en los que pueden ocurrir las transiciones, se añaden dos puertas AND delante de las NOR junto con una entrada adicional de los pulsos de un reloj. Así, las entradas a las puertas NOR sólo estarán activas cuando el pulso de reloj esté en alta y por eso se llama a este circuito *biestable R-S sincronizado a niveles*. Veremos después otros circuitos que todavía acotan más los intervalos de transiciones permitidas (disparo por flancos), donde las señales de control sólo están facilitadas durante las transiciones de los pulsos de reloj (de



alta a baja o de baja a alta).

La figura 8.14 muestra el circuito R-S sincronizado a niveles, un ejemplo de cronograma donde se señalan los intervalos en los que  $R$  y  $S$  están activas y la tabla de verdad correspondiente. Obsérvese que cuando el reloj está en baja ( $Ck = 0$ ) es irrelevante el valor de  $R$  y  $S$  y el biestable permanece en el estado anterior ( $Q_{n+1} = Q_n$ ). En cambio, cuando el pulso de reloj está en alta ( $Ck = 1$ ), las señales externas ( $R$  y  $S$ ) llegan tal cual a las puertas NOR ( $R' = R$ ,  $S' = S$ ) y el biestable funciona de forma normal (recordando la necesidad de no usar la configuración  $R = S = 1$ )

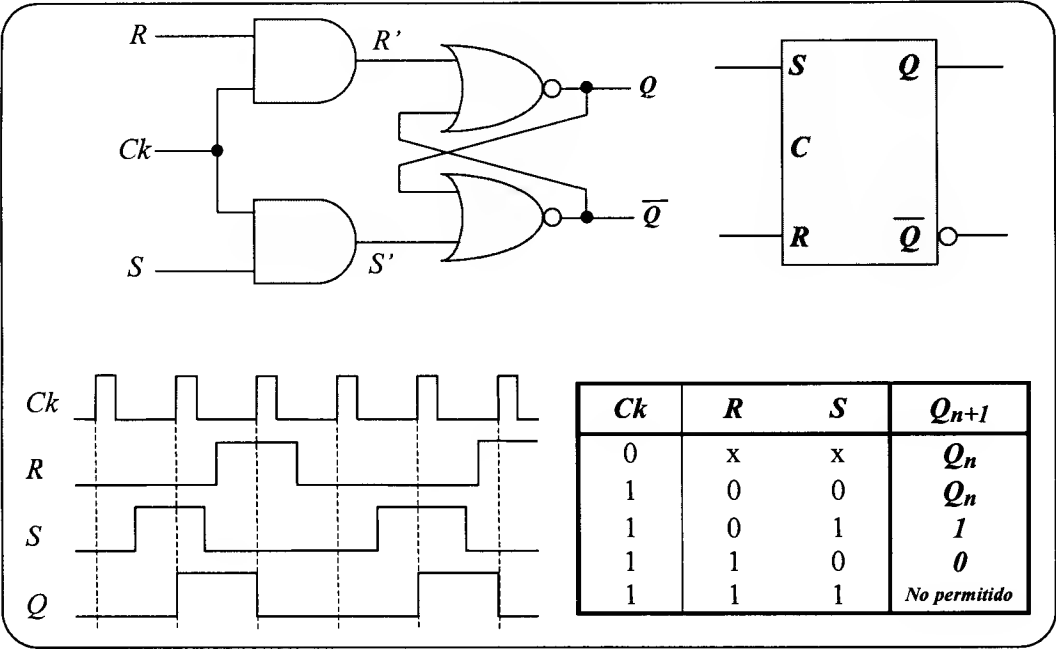


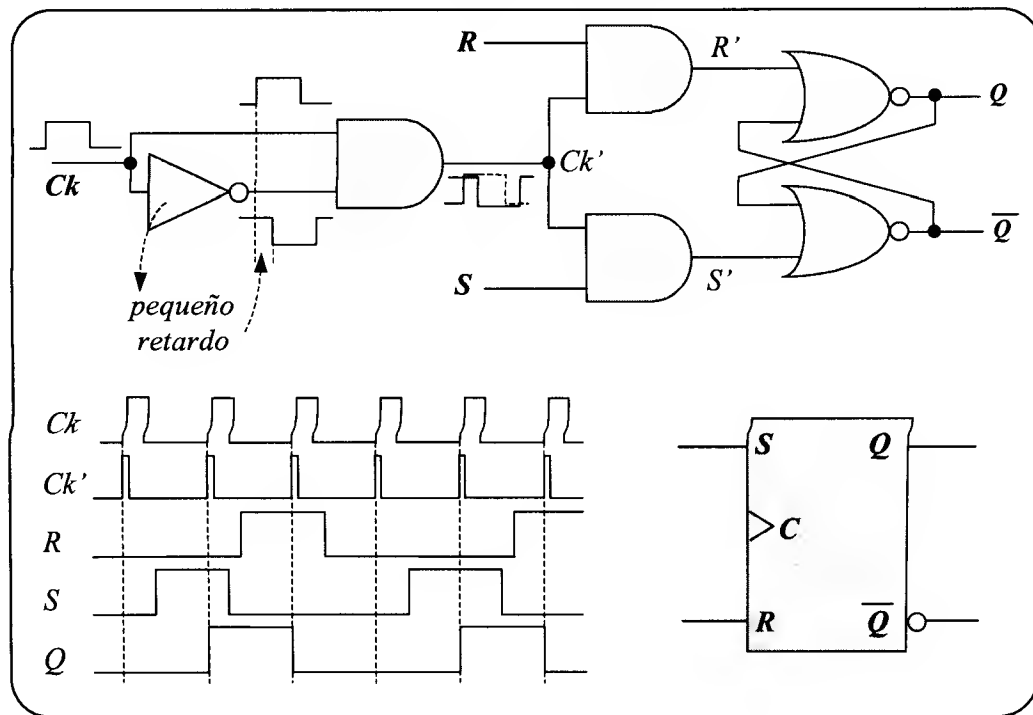
Figura 8.14. Biestable R-S con puertas NOR sincronizado a niveles.

### 8.3.3. Disparo por Flancos

Si el pulso de reloj es muy ancho es posible que se cometan errores debidos a cambios en el valor de  $R$  ó  $S$  durante el intervalo de tiempo en el cual el pulso de reloj está en alta. Para resolver este problema se procura que el pulso de reloj sea muy estrecho y, además, se usan sólo sus transiciones de baja a alta (o de alta a baja) para definir el instante en el que se deja actuar a las variables  $R$  y  $S$ . Decimos entonces que el biestable R-S está disparado por flancos (positivos, “subidas” o negativos,

“bajadas”). Así, como  $R$  y  $S$  son asíncronas, lo usual es que los correspondientes niveles de tensión estén “esperando” a que el pulso de reloj realice una transición para ser operativas, disminuyendo así la probabilidad de que justo en esa transición se produzca de forma no deseada otra transición en  $R$  o  $S$ .

La *figura 8.15* muestra una forma sencilla de producir un pulso muy estrecho asociado a las transiciones del pulso de reloj. La señal de reloj se pasa por un inversor y, ambas, la propia señal y su inversa, se llevan a una puerta AND. Como toda puerta lleva asociado un retardo de transmisión, a la salida de la puerta AND aparece un pulso muy estrecho, justo del valor de ese retardo, Si no fuera suficiente para provocar el disparo de las otras puertas siempre podríamos aumentar el retardo añadiendo más inversores, en número impar. En la *figura 8.15* se incluye también el símbolo de estos biestables, con el triángulo en la entrada de reloj, indicando que es una entrada dinámica.



**Figura 8.15.** Circuito de modificación de los pulsos del reloj para disparar por flancos positivos al biestable R-S.

---

**Ejercicio.** Dibujar el cronograma correspondiente a una misma sucesión de niveles de tensión en las entradas de set ( $S$ ) y reset ( $R$ ) de tres biestables  $R$ - $S$ : a) El básico, b) uno sincronizado a niveles y c) otro disparado por flancos positivos. Dibujar en los tres casos la evolución temporal de la salida  $Q$ .

**Solución:**

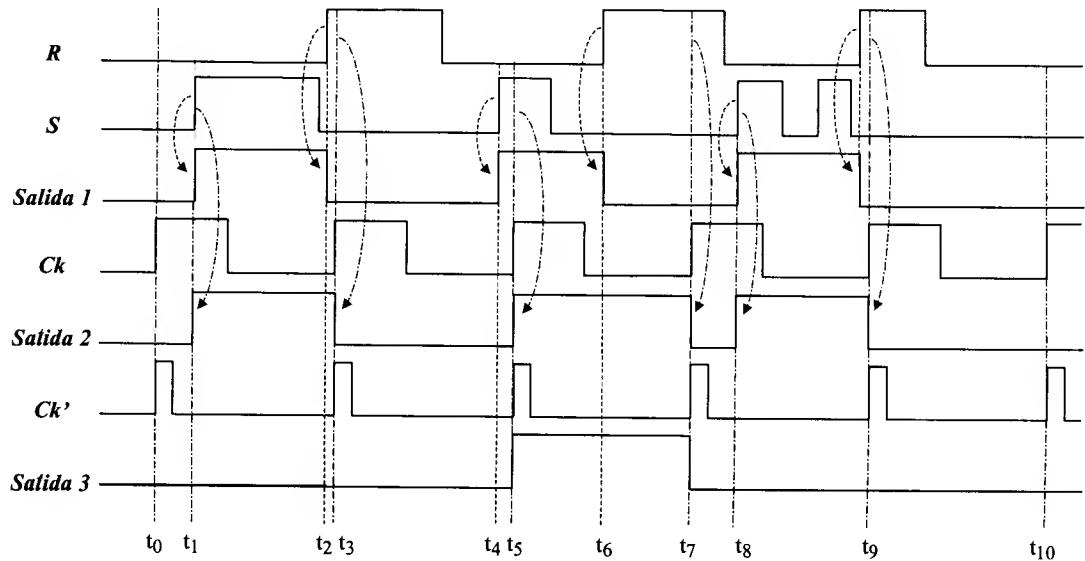
En todos los casos vamos a evitar la configuración  $R=S=1$

**Caso 1: Biestable  $R$ - $S$  Básico.** En este caso la respuesta del circuito es la correspondiente a la tabla de funcionamiento de la *figura 8.11* y el biestable cambiará o no de estado de acuerdo con los cambios en los valores de los niveles de las entradas. Así, para las entradas  $R$  y  $S$  que se muestran en la *figura 8.16* obtenemos como respuesta la *Salida 1*. Para construir esta salida suponemos que inicialmente ( $t=t_0$ ) el biestable está en el estado “0”, es decir,  $Q=0$ . Como inicialmente las entradas son  $R=S=0$  el biestable no cambiará de estado y por tanto  $Q=0$ . En  $t=t_1$  la señal  $S$  pasa a alta dando lugar a la configuración de entrada  $R=0, S=1$  que hace que el biestable cambie de estado y pase al estado de alta, ya que  $S=1$ . A continuación hay un intervalo de tiempo muy estrecho (justo antes de  $t_2$ ) en el que el pulso de  $S$  pasa a baja y aún no ha subido la señal de  $R$ . Durante este tiempo la configuración de entrada es la 00 que como hemos visto en la tabla de funcionamiento del biestable permanece en el mismo estado anterior. En  $t_3$  es la entrada  $R$  la que pasa a alta mientras que  $S$  permanece en 0, por tanto, esta configuración pasará al biestable al estado de baja. En este estado permanece hasta que vuelve a aparecer un pulso en  $S$  que hace que vuelva a pasar a alta al biestable. Y así sucesivamente.

Si observamos globalmente la salida 1 vemos que cada vez que hay un paso de baja a alta en la entrada  $S$  (con  $R=0$ ) la *salida 1* también pasa de baja a alta y pasa de alta a baja cuando la entrada  $R$  lo hace de baja a alta (con  $S=0$ ). Sin embargo, entre  $t_8$  y  $t_9$  la señal  $S$  presenta dos pulsos, pero como durante todo ese tiempo  $R$  está en baja, el resultado es el mismo que si sólo existiera el primero de los pulsos (ante la configuración 00 permanece en el mismo estado o sea, en alta y ante la configuración  $R=0$  y  $S=1$  pasa a alta, o sea en el estado en el que ya estaba).

**Caso 2: Biestable  $R$ - $S$  sincronizado a niveles.** La respuesta de un biestable  $R$ - $S$  sincronizado a niveles (circuito de la *figura 8.14*) ante las mismas señales de entrada del caso anterior es la *salida 2* que se muestra en la *figura 8.16*.

El primer cambio tiene lugar en  $t_1$  ya que la entrada  $S$  pasa a alta y el reloj está también en alta. Entre  $t_1$  y  $t_2$  el reloj cambia de estado pero la salida no lo hace hasta  $t_3$  que pasa a baja, ya que se vuelve a presentar la condición de que el reloj está en alta y las señales de entrada presentan la configuración  $R=1$  y  $S=0$ . Los siguientes cambios se realizan siempre que el reloj pasa a alta y en las entradas están las configuraciones 01 ó 10 ( $t_5, t_7$  y  $t_9$ ).



**Figura 8.16.** Respuestas de los biestables R-S básico, sincronizado a niveles y disparado por flancos positivos a las mismas señales de entrada.

**Caso 3: Biestable R-S disparado por flancos positivos.** El circuito es el de la figura 8.15 y la salida que se genera ante las mismas señales de entrada de los casos anteriores es la que se muestra como **salida 3** de la figura 8.16. En este caso, como vimos en la teoría, a partir del reloj externo,  $Ck$ , se genera una nueva señal de reloj ( $Ck'$ ) de igual periodo pero con los pulsos mucho más estrechos y que se corresponden prácticamente con los flancos de subida. Estos pulsos aparecen en  $t_0, t_3, t_5, t_7, t_9$  y  $t_{10}$ .

En  $t_1$  la salida es 0 puesto que cuando el reloj pasa a alta la configuración de entrada es 00. En  $t_3$  la salida sigue siendo 0 porque en este caso la entrada que está en alta es  $R$  y su función es poner a 0 al biestable. El primer cambio tiene lugar en el instante  $t_5$  en el que el reloj pasa a alta y la configuración de entrada es  $R=0$  y  $S=1$ , por tanto, el biestable pasa al estado de alta. Permanece en este estado hasta el siguiente pulso de reloj,  $t_7$ , en el que la configuración de entrada ya ha cambiado y

en ese instante presenta los valores  $S=0$  y  $R=1$ , por lo que el biestable pasa al estado de baja. En  $t_9$  y  $t_{10}$  sigue en estado de baja porque las configuraciones de entrada son  $S=0$  y  $R=1$  en  $t_9$  y  $S=R=0$  en  $t_{10}$ .

Como podemos observar los tres biestables tiene distinto comportamiento y ante las mismas entradas dan lugar a salidas diferentes.

### 8.3.4. R-S Sincronizado a Nivel y con Entradas Asíncronas de Preset y Clear

En la anterior configuración del R-S los cambios de estado no podían producirse más que a través de las entradas  $R$  y  $S$  y de forma síncrona, coincidiendo con el estado de alta de los pulsos del reloj. Sin embargo, para conseguir una mayor flexibilidad en el diseño de otros circuitos tales como contadores o registros de desplazamiento, para establecer condiciones iniciales y para provocar cambios forzados de forma asíncrona, sin necesidad de coincidencia con los pulsos de reloj, se incluyen dos nuevas entradas asíncronas adicionales llamadas de *preset* (puesta a 1) y *clear* (puesta a cero). Esto se puede conseguir añadiendo dos puertas OR detrás de las NOR y cerrando ahora los lazos de realimentación desde estas puertas OR adicionales, tal como se muestra en el circuito de la figura 8.17.

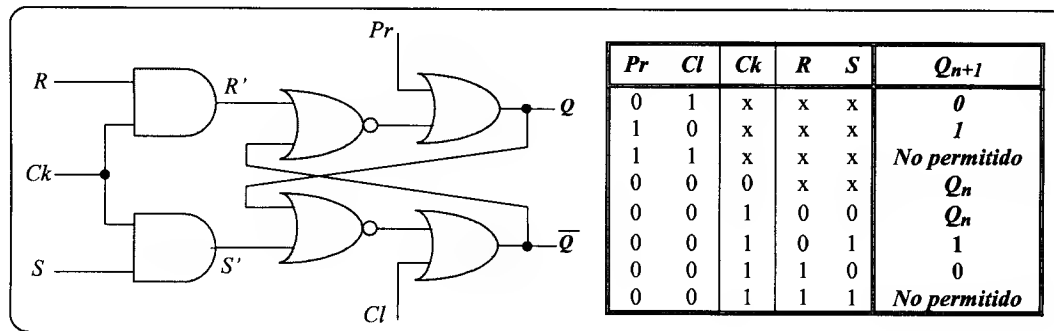


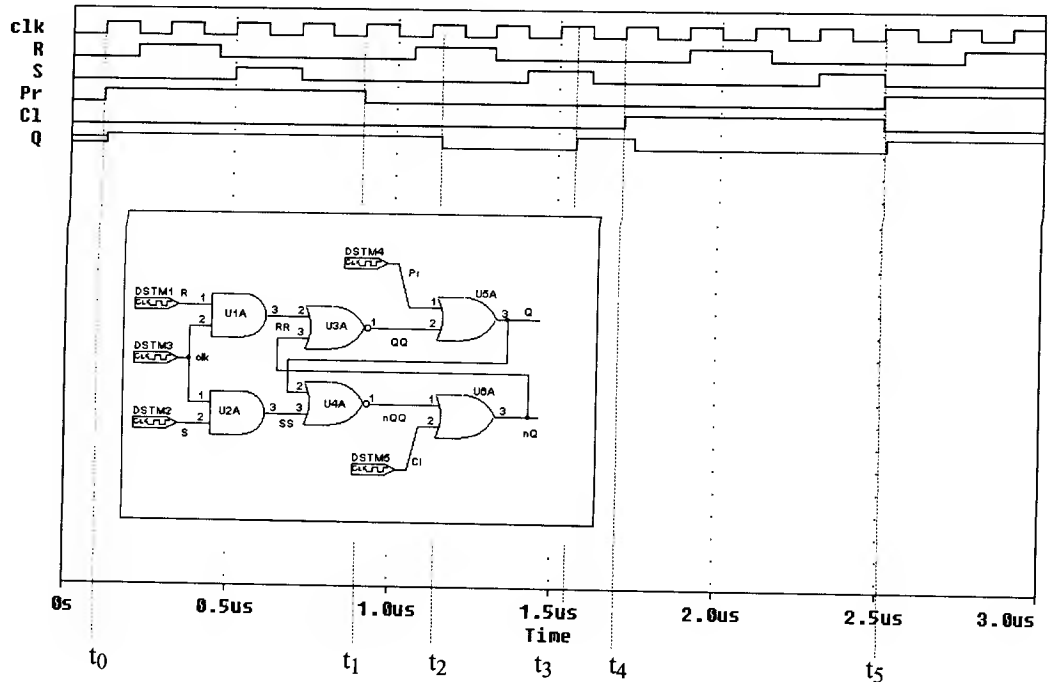
Figura 8.17. Biestable R-S sincronizado a nivel con entradas  $Pr$  y  $Cl$  asíncronas.

Así, independientemente del valor de las entradas  $R$  y  $S$  y del reloj y, actuando de forma prioritaria, siempre que preset pase a alta ( $Pr = 1$ ), la salida del biestable pasará también a alta ( $Q=1$ ). Inversamente, si clear pasa a alta ( $Cl = 1$ ), la salida pasará a baja ( $Q=0$ ), independientemente también

del valor de las otras señales y con prioridad.

**Ejercicio:** Simular el biestable R-S sincronizado a nivel con preset y clear de la figura 8.17.

**Solución:**



**Figura 8.18.** Simulación de un biestable R-S controlado por las señales de preset (*Pr*) y clear (*Cl*).

En el cronograma de la simulación podemos observar cómo actúan el preset (*Pr*) y el clear (*Cl*). Inicialmente, en  $t_0$ , *Pr* pasa a 1 mientras que *Cl* permanece en 0, por tanto la salida es 1 independientemente de los valores de las entradas *R* y *S*. Posteriormente la señal *Pr* pasa a 0 presentándose la configuración de 00 para *Pr* y *Cl*. Con esta configuración (que dura desde  $t_1$  hasta  $t_4$  en que *Cl* pasa a alta) el biestable R-S actúa como tal, produciéndose la salida de acuerdo con las distintas configuraciones de *R* y *S* y cambiando de estado cuando el reloj está en alta. Obsérvese los retardos (de 27 y 55ns) existentes entre los cambios de estado en la salida, *Q*, y las subidas de reloj debidos a la suma de los retardos producidos por las distintas puertas

En  $t_4$  la señal  $Cl$  pasa a alta haciendo que la salida pase a baja. Por último en  $t_5$  la señal  $Pr$  pasa a alta poniendo al biestable en estado de alta.

---

## 8.4. BIESTABLES J-K

Un problema importante de los biestables R-S es su ambigüedad ante la configuración  $R=S=1$ , que hacía que la salida  $Q$  fuera igual a la  $\bar{Q}$ . Este problema lo resuelve la configuración J-K haciendo que en este caso ( $R=S=1$ ) el biestable cambie de estado ( $Q_{n+1} = \bar{Q}_n$ ). Así queda perfectamente definido el comportamiento para las cuatro configuraciones posibles en las señales de control. La entrada  $J$  hace la función de la  $S$  (“Set”, puesta a 1) y la  $K$  la de la  $R$  (“Reset”, puesta a 0), de forma que ahora, cuando no hay pulso de reloj no hay cambios, pero cuando al pulso de reloj está en alta actúan las entradas  $J$  y  $K$ . Así,

Si	$J=K=0$	→	No cambia de estado	→	$Q_{n+1} = Q_n$
Si	$J=1$ y $K=0$	→	Pasa a “1”	→	$Q_{n+1} = 1$
Si	$J=0$ y $K=1$	→	Pasa a “0”	→	$Q_{n+1} = 0$
Si	$J=K=1$	→	Cambia de estado	→	$Q_{n+1} = \bar{Q}_n$

La *figura 8.19* muestra el circuito correspondiente al biestable J-K sincronizado a nivel. La clave del diseño está en las realimentaciones que se traen de las salidas de las puertas NOR a las puertas AND de entrada para producir las señales efectivas de *set* y *reset* ( $S'$  y  $R'$ ). Así se evita el problema del R-S, porque cuando las entradas  $J$  y  $K$  están a la vez en alta, la salida de la puerta OR que está activa en ese momento ( $Q$  y  $\bar{Q}$ ) hace que la puerta AND correspondiente (la de  $K$  ó la de  $J$ ) pase a alta su salida ( $R'$  ó  $S'$ ) y, por consiguiente, haga volcar al biestable en cualquier caso. En efecto, como

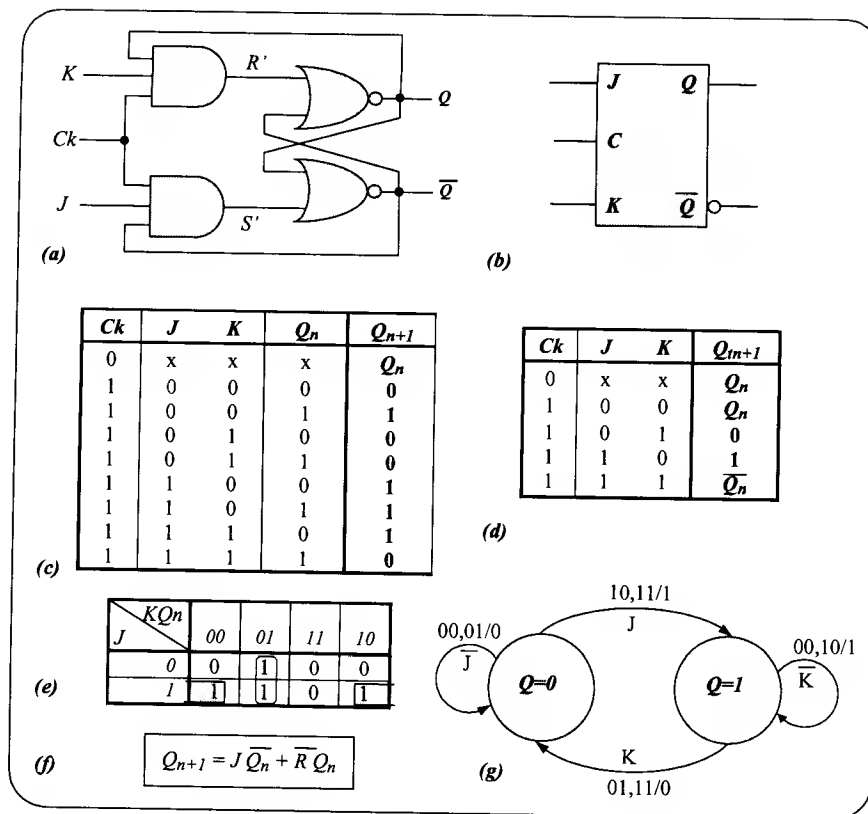
$$R' = Q_n \cdot K \cdot Ck$$

$$S' = \bar{Q}_n \cdot J \cdot Ck$$

[8.11]

Si  $Q_n = 1$ ,  $R' = 1$  y  $Q_{n+1} = 0$

Análogamente si  $Q_n = 0, \bar{Q}_n = 1, S' = 1$  y  $Q_{n+1} = 1$



**Figura 8.19.** Biestable J-K sincronizado a niveles. (a) Circuito. (b) Símbolo lógico. (c) Tabla de verdad completa. (d) Tabla simplificada. (e) Minimización de la expresión de  $Q_{n+1}$ . (f) Ecuación lógica resultante. (g) Diagrama de transición de estados.

Obsérvese que si dejamos permanentemente  $J=K=1$ , hay un cambio de estado cada pulso de reloj. Este comportamiento del J-K corresponde a la configuración de disparo T ("Toggle") de la que hablaremos más adelante al mostrar cómo se pueden sintetizar todos los tipos de biestables a partir del J-K.

En la figura 8.19.a se muestra la tabla de verdad ampliada, haciendo aparecer las ocho configuraciones lógicas posibles para  $J$ ,  $K$  y  $Q_n$  y viendo los términos mínimos que definen el nuevo estado,  $Q_{n+1}$ . En 8.19.b se minimiza y en 8.19.c se muestra el resultado de la minimización con la ecuación que define el comportamiento y que repetimos aquí por comodidad



$$Q_{n+1} = Q_n (\overline{JK} + JK) + \overline{Q}_n (J\overline{K} + JK) = Q_n \overline{K} + \overline{Q}_n J \quad [8.12]$$

Finalmente, en la parte (d) de la *figura 8.19* se muestra el diagrama de transición de estados.

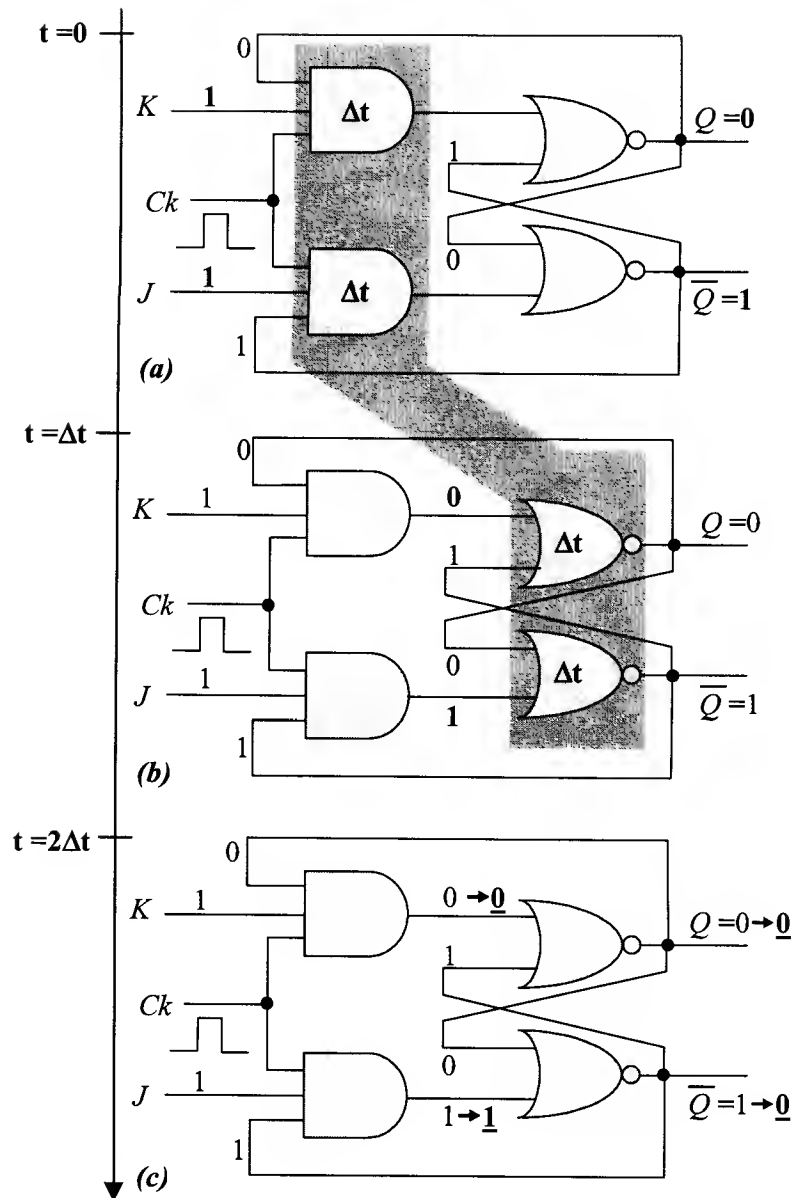
Aunque los biestables J-K disparados por nivel resuelven el problema de los R-S cuando  $R=S=1$ , todavía quedan otros problemas sin resolver que nos van a llevar a configuraciones más complejas, con dos biestables R-S disparados por flancos y con separación de la orden de disparo de su ejecución. El problema fundamental del biestable J-K es que cuando  $J=K=1$  y el pulso de reloj está en alta se produce una situación inestable, con la salida oscilando entre “0” y “1” como consecuencia de los retardos de propagación de las señales a través de los dos lazos de realimentación que se cierran sobre las puertas NOR (como en el R-S) y sobre las puertas AND de entrada.

Veamos este problema analizando con detalle la configuración  $J=K=1$  a partir del estado inicial  $Q=0$  y suponiendo que durante el tiempo al que se hace referencia en el proceso de análisis la señal del reloj permanece en alta. La *figura 8.20* ilustra la secuencia de procesos internos en seis “fotos”, instantáneas, del proceso de realimentación y transmisión de las señales. Suponemos que ambas puertas, las AND y las NOR, poseen el mismo retardo de transmisión ( $\Delta t$ ) y que la transmisión por conductores es instantánea.

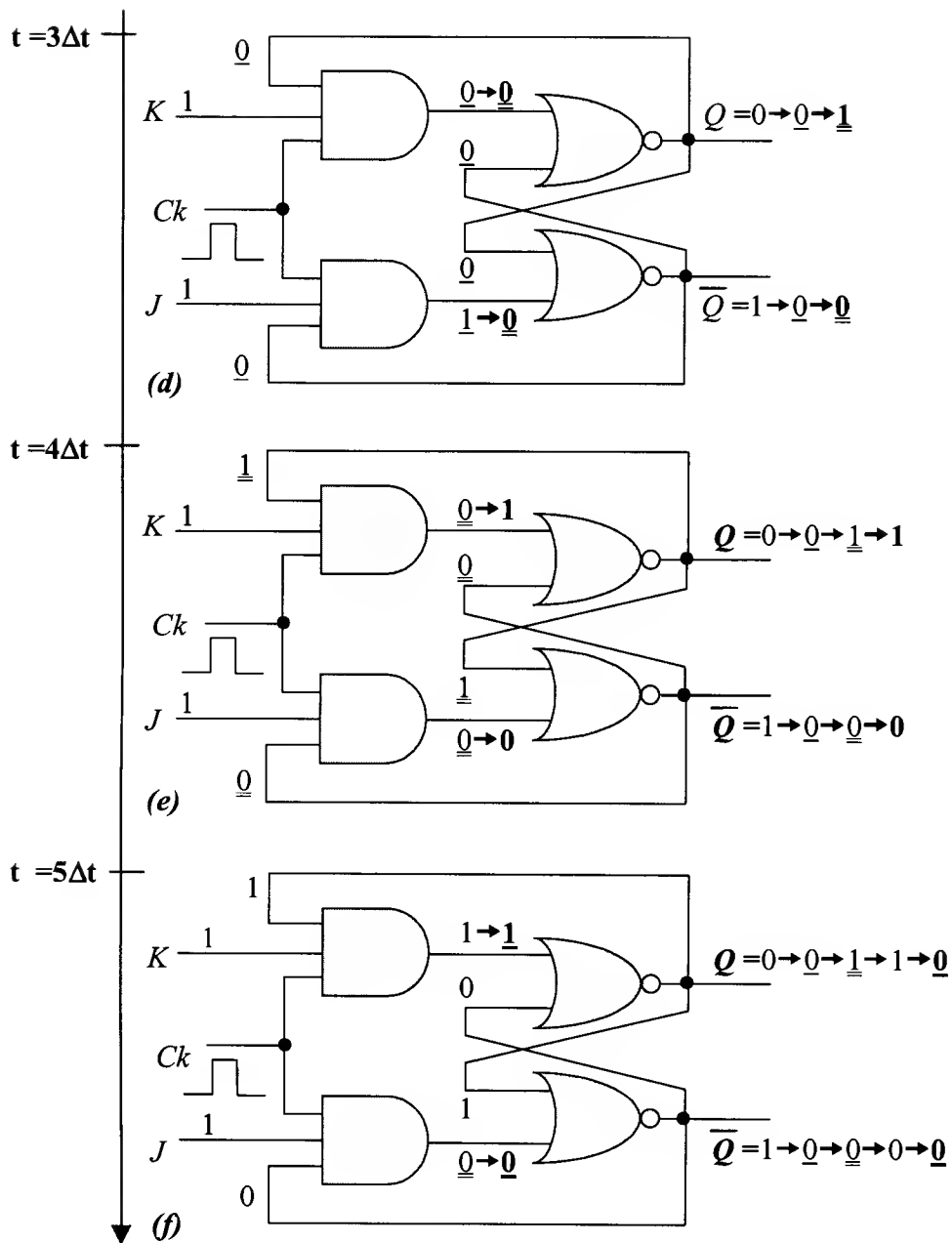
En la primera “foto” (parte (a) de la *figura 8.20*) tenemos las condiciones iniciales. Si hemos supuesto que  $J=K=1$ ,  $Q=0$  y  $\overline{Q}=1$ , y el reloj en alta, estos valores de  $Q$  y  $\overline{Q}$  aparecen instantáneamente a las entradas de las puertas AND y NOR. Así, inicialmente tenemos  $Q=0$ ,  $K=1$  y  $Ck=1$  como entrada de la primera NAND y  $\overline{Q}=1$ ,  $J=1$  y  $Ck=1$  como entrada de la otra NAND. De las NOR (en este momento) sólo sabemos que  $\overline{Q}=1$  se realimenta a la primera NOR y  $Q=0$  se realimenta a la segunda NOR.

Un intervalo de tiempo,  $\Delta t$ , más tarde (2ª “foto”, correspondiente a la parte (b) de la *figura 8.20*), ambas puertas AND ya han calculado y tenemos en sus salidas los resultados de esos cálculos. Así, en la salida de la AND de  $K$  tenemos un “0” y en la AND de  $J$  un “1”, que se transmiten a la entrada de sus NOR correspondientes.

Un intervalo de tiempo más tarde ( $t=2\Delta t$ ), ya han calculado también las puertas NOR (las AND lo han hecho concurrentemente) y nos han producido las salidas  $Q=0$  (NOR de 0 y 1) y  $\overline{Q}=0$  (de nuevo, NOR de 0 y 1).



**Figura 8.20.** Análisis detallado de la evolución temporal del proceso de realimentación en un biestable J-K disparado por pulsos de reloj, sincronizados al nivel de alta. Se muestran 8 “fotos” (instantáneas) correspondientes a las distintas fases de transmisión de las señales en los lazos de realimentación a través de los retardos ( $\Delta t$ ) inherentes a las puertas AND y NOR.



**Figura 8.20. (cont)** Para facilitar la comprensión de la figura hemos señalado con **negrita** los valores calculados en cada intervalo de tiempo y hemos usado el código de distintos subrayados para seguir la trayectoria y la secuencia de los sucesivos valores de  $Q$  y  $\bar{Q}$ .

Este estado externo se realimenta de nuevo a la entrada de las AND, de forma que en la cuarta “foto” ( $t=3\Delta t$ ), si todavía sigue activo (en alta) el pulso del reloj y los valores de  $J=K=1$ , el circuito realimenta este estado transitorio y da lugar a una salida  $Q=1$  y  $\bar{Q}=0$ . Este proceso se repite dando lugar a una secuencia oscilatoria no deseada. Si se sigue este análisis, no es difícil de comprobar que la secuencia de salida es (0,1), (0,0), (1,0), (1,0), (0,0), (0,1), etc...

Una primera solución de este problema es disminuir la anchura del pulso de reloj para que no de tiempo más que a una transición. Es decir, que sea del orden de los retardos de propagación. Así, cuando la realimentación presente un nuevo valor de  $Q$  y  $\bar{Q}$  a las entradas de las puertas AND, el pulso del reloj ya está en baja y, por consiguiente, no son operativas. El estado inicial no debe poder provocar más que una transición (en su caso).

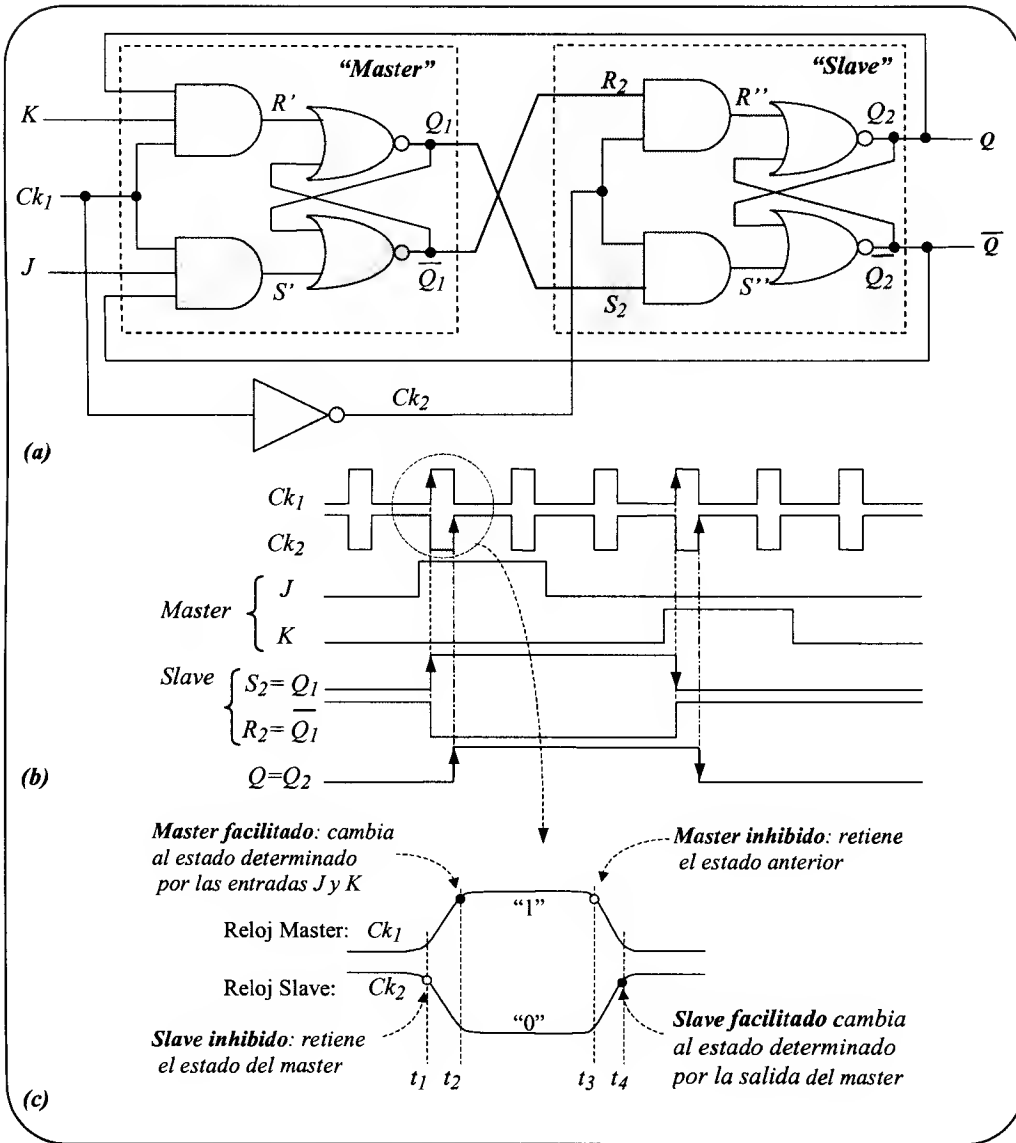
#### 8.4.1. Configuración “Master-Slave”

Para evitar estos problemas de los biestables J-K disparados a nivel, se usa una configuración con dos biestables conectados en serie y con relojes complementarios que interrumpen la conexión lógica entre la salida y la entrada. Es decir, entre la generación de la orden de disparo (función de las señales de control  $J$ ,  $K$  y  $Ck$ ) y su ejecución (función de la salida del primer biestable), que opera en la fase complementaria del reloj. Así, se rompe la parte interna del lazo de realimentación.

Esta configuración del biestable J-K se llama “Master-Slave” (maestro-esclavo)<sup>(\*)</sup> y se muestra en la *figura 8.21*. Al maestro le entra directamente el pulso de reloj y las señales  $J$  y  $K$ . Al esclavo le entra las salidas del primero y el pulso de reloj invertido de forma que nunca pueden estar ambas secciones facilitadas a la vez. Cuando el pulso de reloj realiza una transición, el maestro captura los datos presentes en las entradas  $J$  y  $K$ . A continuación se inhibe y queda aislado tanto de las entradas como de las realimentaciones de la señal de salida ( $Q$  y  $\bar{Q}$  del esclavo). La desconexión del maestro coincide con la conexión del esclavo de forma que cualquier cambio que pueda presentarse en la entrada ya no afectará a la salida.

---

(\*) “Master-Slave” ha sido traducido por “maestro-esclavo”, “dueño-esclavo”, “principal-subordinado”, intentando transmitir en todos los casos la idea de separación y subordinación de funciones



**Figura 8.21.** Configuración J-K "master-slave". (a) Circuito. (b) Cronograma con la evolución temporal de las señales durante el disparo de un J-K "master-slave". (c) Reloj en  $Ck_1$  y  $Ck_2$ , ilustrando los instantes de conmutación en ambas secciones.

En la parte (b) de la figura 8.21 se muestra el cronograma ilustrando los puntos en los que se realizan las transiciones en los pulsos de reloj y los correspondientes cambios de estado en las

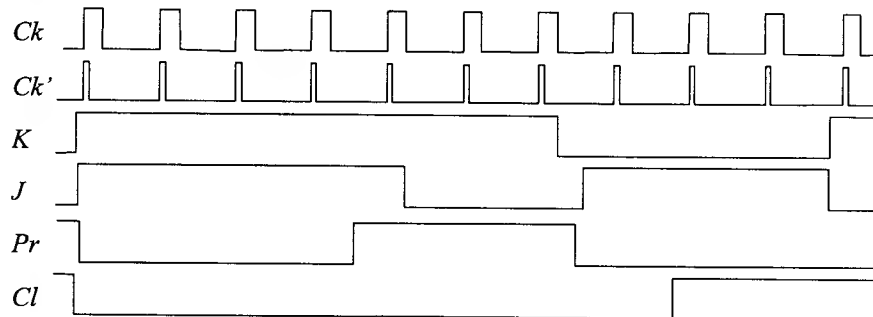
secciones maestro y esclavo del biestable durante el proceso de disparo. En 8.21.c se muestra el detalle de estas transiciones sobre el pulso ( $Ck_1$ ) y su complementario ( $Ck_2$ ).

Obsérvese que el resto de la configuración del J-K se mantiene igual que antes. Es decir, continúan existiendo las dos realimentaciones externas desde  $Q$  a  $K$  y desde  $\bar{Q}$  a  $J$ .

Existen otras soluciones alternativas a la configuración "master-slave" que mantienen sus características de aislamiento pero son más rápidas. En la configuración "master-slave" hay un retardo intrínseco a su modo de funcionamiento pues hay que disparar dos biestables en serie para completar una transmisión desde las entradas  $R$  y  $S$  del "master" hasta las salidas ( $Q$ ,  $\bar{Q}$ ) del "slave". Para resolver este problema se han diseñado otros tipos de biestables disparados por flancos ("edge-triggered"), tipo SN7470, 101, 102 ó 108 (dual). La señal de salida de un biestable disparado por flancos aparece prácticamente a continuación de la señal de disparo. El único retardo es el inherente a las puertas NAND ó NOR con las que se ha diseñado.

---

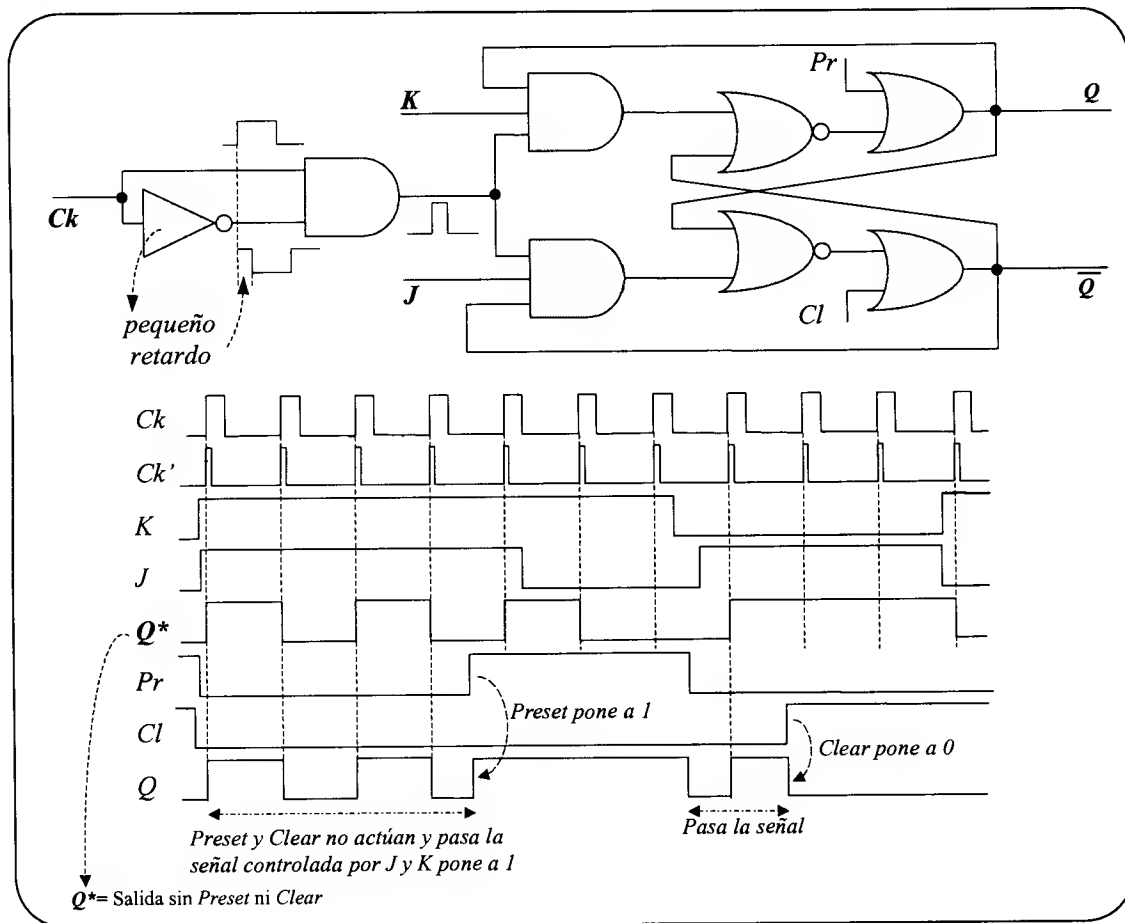
**Ejercicio** *El biestable J-K disparado a nivel admite las mismas mejoras que hemos comentado previamente para el R-S. Es decir, el disparo por flancos y la inclusión de terminales asíncronos para preset y clear. Explicar el funcionamiento del J-K disparado por flancos con entradas asíncronas de preset y clear y calcular la evolución temporal de la salida  $Q$  para el conjunto de señales  $Pr$ ,  $Cl$ ,  $J$  y  $K$  que se muestran en el cronograma de la figura 8.22.*



**Figura 8.22.** Cronograma del J-K disparado por flancos con  $Pr$  y  $Cl$ .

## Solución

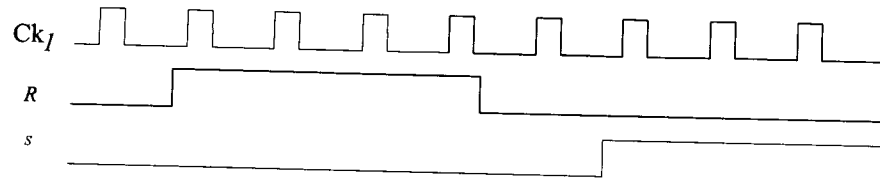
Para el disparo por pequeños pulsos asociados a las transiciones de baja a alta del reloj, podemos usar un circuito como el de la *figura 8.23* y para incluir *preset* y *clear* asíncronos ponemos dos nuevas puertas OR a la salida de las NOR, con lo que el circuito queda:



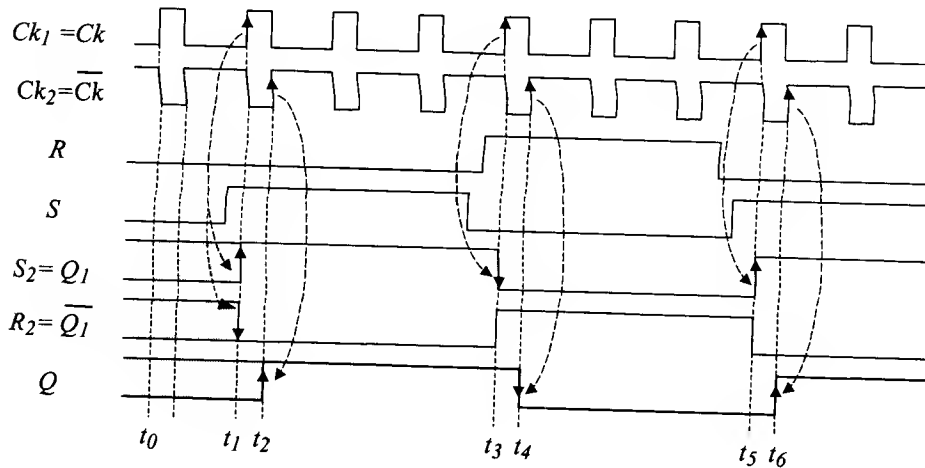
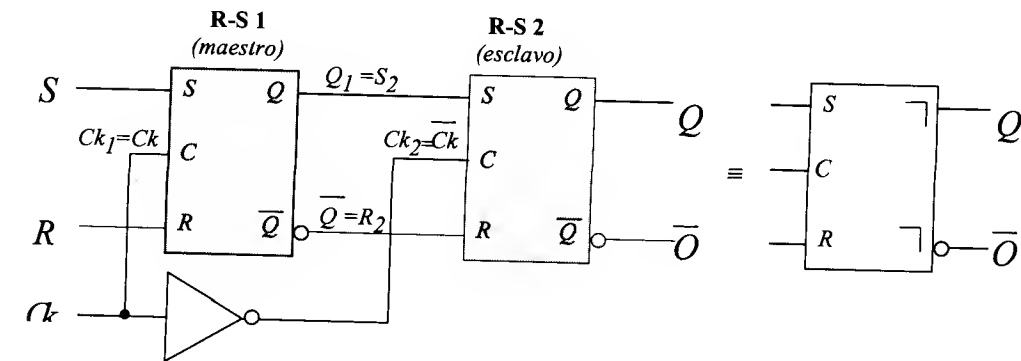
**Figura 8.23.** Circuito de disparo del J-K por pulsos estrechos.

**Ejercicio:** La configuración "esclavo-maestro" que hemos visto para el biestable J-K puede usarse para cualquier otro biestable. Consiste en usar dos biestables iguales uno para generar la orden de disparo y otro para ejecutarla y almacenar el estado en un intervalo

complementario de los pulsos del reloj usados en el primero. Proponer un circuito para biestables R-S con puertas NOR y completar el cronograma correspondiente al conjunto de señales significativas ( $R$ ,  $S$ ,  $Ck$ ,  $Q_1$ ,  $\bar{Q}_1$ ,  $Q_2$ ,  $\bar{Q}_2$ ).



**Solución:**



**Figura 8.24.** Configuración maestro-esclavo con R-S



Hay soluciones integradas en TTL y en otras tecnologías para todos los biestables J-K descritos en este apartado (disparados por flancos o por pulsos, en configuración maestro-esclavo y con Pr y Cl. Por ejemplo el circuito SN7473 contiene dos biestables J-K “master slave” independientes, disparados por pulsos y con salidas complementarias. El 74109 contiene también dos biestables J-K independientes disparados por flancos positivos y con entradas asíncronas de *preset* y *clear*, activas en baja. Un nivel de tensión bajo en estas entradas pone a “1” ó a “0”, respectivamente, las salidas del biestable. Cuando las entradas de *preset* y *clear* están inactivas (en alta), las señales de control en *J* y *K* producen las transiciones correspondientes en las salidas en los flancos positivos del pulso de reloj. Como veremos en el siguiente apartado, estos circuitos pueden configurarse para operar también como T o como D. La versión de uso libre del simulador PSpice tiene también muchas de estas opciones (7474, 7478, 74109, etc.)

## 8.5. BIESTABLES T Y D

El biestable T cambia de estado ante cada pulso de reloj. Es el ejemplo que usamos para introducir la teoría de autómatas al comienzo del tema. Se obtiene, por ejemplo, a partir de un J-K conectando ambas entradas juntas. La *figura 8.25* muestra la conversión de J-K en T, la tabla de verdad y el diagrama de transición de estados.

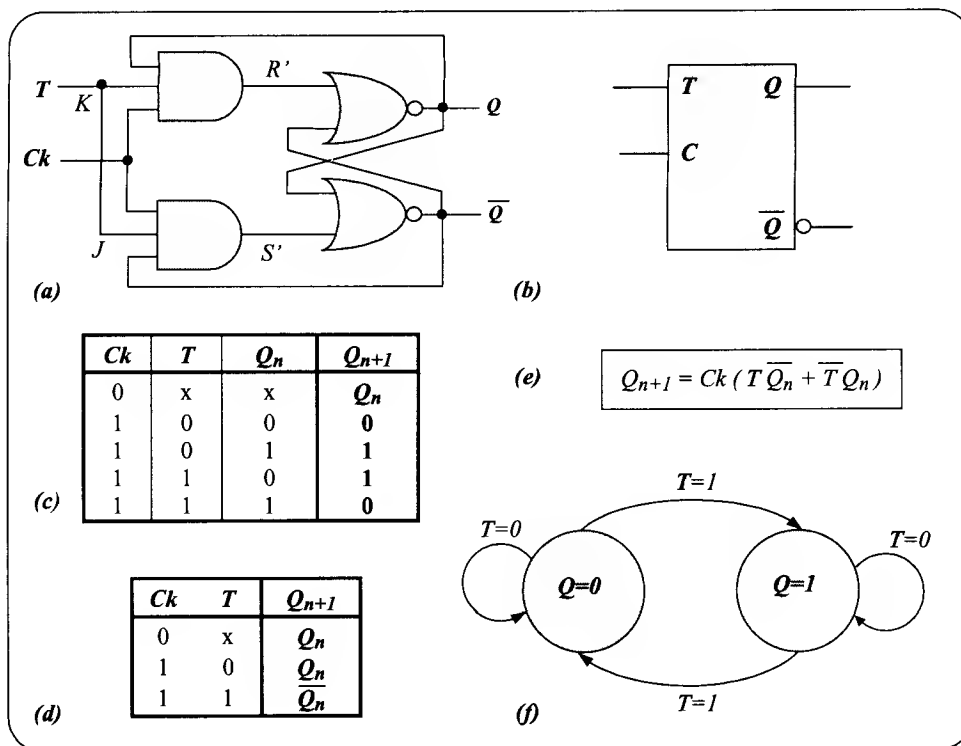
La ecuación lógica de este biestable se puede escribir por simple inspección de los dos términos mínimos que posee  $Q_{n+1}$  en función de  $T$  y  $Q_n$ .

$$Q_{n+1} = T\bar{Q}_n + \bar{T}Q_n \quad [8.13]$$

Es decir, el biestable T genera un estado de salida que, es el "OR-exclusivo" de la entrada y el estado anterior. Se dispara siempre que no hay coincidencia entre el estado actual ( $Q_n$ ) y la entrada ( $T$ ). Ante un tren de impulsos en su entrada de reloj, cambia de estado cada nuevo pulso.

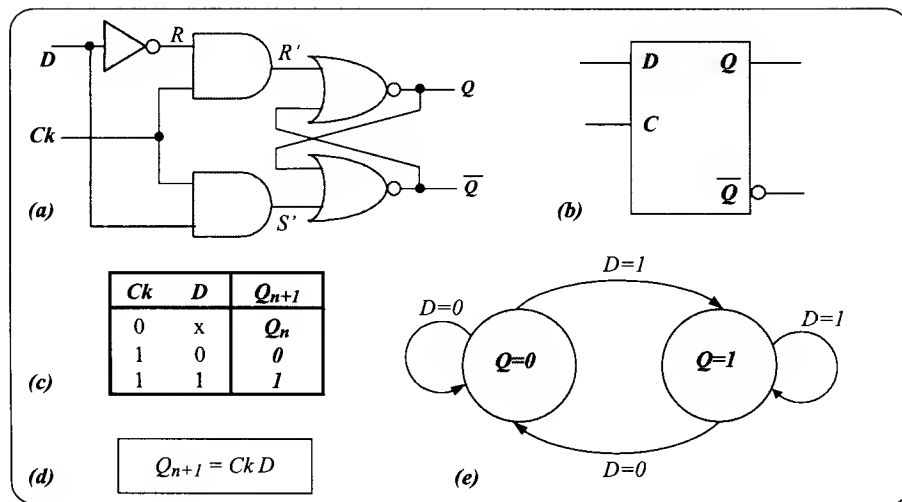
El biestable D (Delay) representa el retardo. Su salida en cada intervalo coincide con la entrada en el intervalo anterior. Al igual que en el T, se puede obtener a partir de un R-S ó un J-K uniendo la entrada *D* directamente a *S* ó *J* y usando un inversor para activar la *R* ó la *K*, tal como se muestra en la *figura 8.26*, donde se incluyen también la tabla de verdad, la ecuación y el diagrama de transición de estados. La línea *D* es la línea de *entrada de datos* al biestable. Cuando  $D=1$ ,  $S=1$  y  $R=0$  y el biestable pasa a alta. Inversamente, cuando  $D=0$ ,  $S=0$  y  $R=1$  y el biestable pasa a baja. Al estar unidas

$R$  y  $S$  a través de un inversor, la configuración  $R=S=1$ , no puede darse nunca. Al estar sincronizado a niveles, el dato ( $D=0$ ,  $D=1$ ) sólo se puede transmitir a la salida durante el pulso de reloj,  $Ck$ .

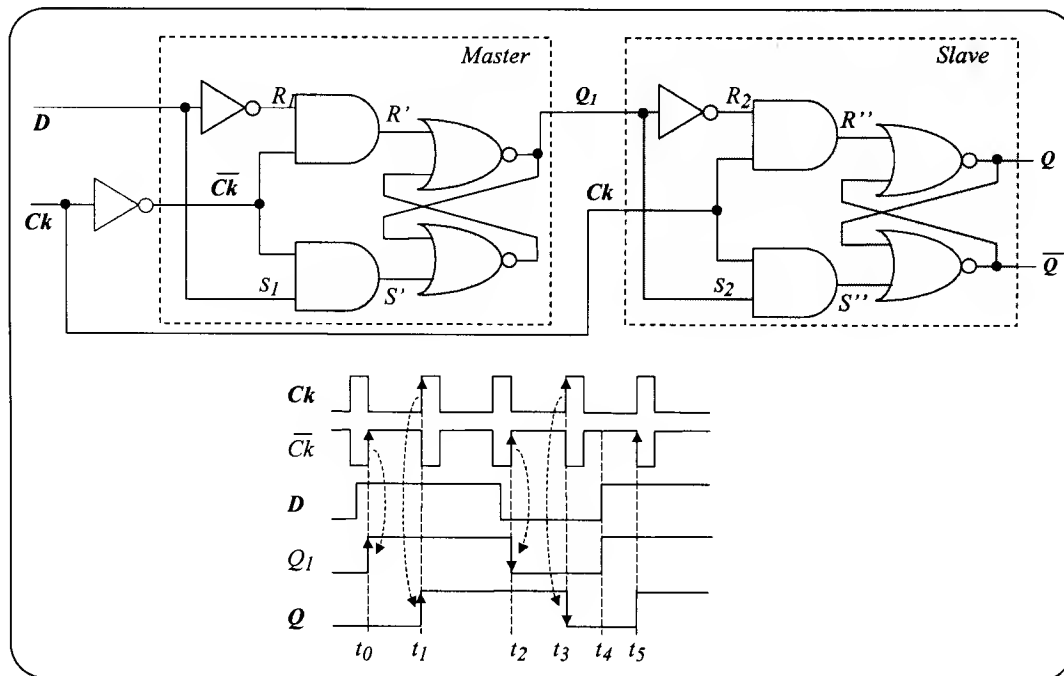


**Figura 8.25** Configuración T. (a) Obtención del T a partir de un J-K haciendo  $T=J=K$ . (b) Símbolo lógico. (c) y (d) Tablas de verdad. (e) Expresión lógica. (f) Diagrama de transición de estados.

Todos los biestables admiten la configuración "master-slave" duplicando el circuito básico correspondiente a su disparo a niveles y usando la señal de reloj y su complementaria para activar ambas secciones de forma sucesiva y mutuamente exclusiva. La *figura 8.27* muestra un ejemplo de biestable D "master-slave" disparado en el flanco positivo de los pulsos de reloj. Obsérvese que esta *figura* es una repetición de la 8.26.a duplicada y con un inversor añadido a la entrada de reloj de la primera etapa. Cuando  $Ck=0$ , antes del primer flanco de subida en el pulso de reloj, la salida  $Q_1$  del primer biestable R-S sigue a la señal de entrada,  $D$ . Cuando  $Ck$  pasa de 0 a 1, el valor de  $Q_1$  se estabiliza pues la sección "master" queda inhibida. Entonces la salida  $Q_2$  del segundo biestable sigue al valor de  $Q_1$ .



**Figura 8.26.** Configuración D. (a) Circuito. (b) Símbolo lógico. (c) Tabla. (d) Ecuación. (e) Diagrama de transición de estados.



**Figura 8.27.** Configuración D "master-slave" y diagrama de tiempos.

### 8.5.1. D disparado por Flancos

Aunque ya hemos descrito un procedimiento general para pasar del disparo por nivel al disparo por flancos de cualquier biestable (retardando el pulso, invirtiéndolo pasándolo por una puerta AND), hay diseños específicos para biestables D disparados por flancos, como el que se muestra en la *figura 8.28*, correspondiente a circuitos tipo SN74LS74A, al que inicialmente y por razones pedagógicas hemos despojado de las entradas de *preset* y *clear*.

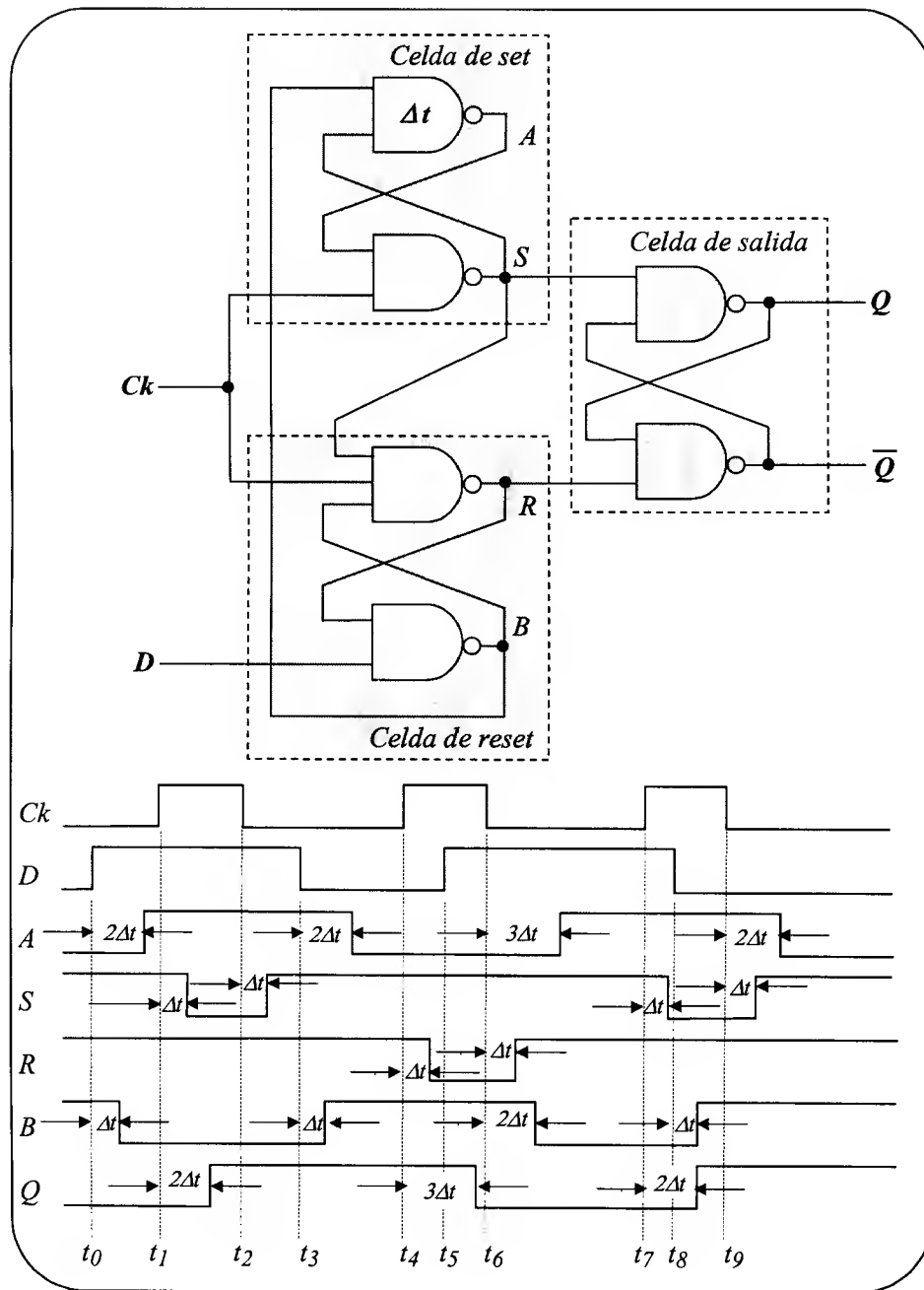
Así, el núcleo de un biestable D disparado por flancos consta de tres celdas R-S (celda de *set*, celda de *reset* y celda de *salida*). Las dos primeras generan las señales internas *R* y *S*, que son recogidas por la última, separando la entrada de la salida.

La celda de *set* sigue los cambios en la señal de reloj si  $D=1$ , en el momento de la transición de baja a alta de *Ck*, mientras que la celda de *reset* sigue al reloj si  $D=0$  en el momento de la transición de baja a alta de *Ck*. Es decir, las señales en *A* y *B* registran los cambios de *D* cuando  $Ck=0$ .

Sigamos ahora el cronograma de la parte (b) de la *figura 8.28*. Mientras que el reloj está en baja ( $Ck=0$ ), las señales *S* y *R* están en alta, ya que son salidas de una puerta NAND con una entrada en baja, reteniendo así el estado que tenía la celda R-S de salida. Por ejemplo, si el estado inicial era ( $Q=0$ ) la puerta de *Q* tiene el NAND de (1,1) que es 0 y la de  $\bar{Q}$  el NAND de (1,0) que es 1.

Cuando, en  $t=t_0$ , *D* pasa a alta, las salida *B* pasa a baja ( $B=0$ ) un tiempo después igual al retardo de transición de la NAND ( $\Delta t$ ). Así, en  $t=t_0+\Delta t$ ,  $B=0$  y otro retardo más tarde, en  $t=t_0+2\Delta t$ , *A* pasa a alta ( $A=1$ ). La salida *Q* no se modifica hasta que la subida del pulso de reloj ( $Ck=1$ ) en  $t=t_1$  fuerza a que *S* pase a cero en  $t=t_1+\Delta t$  y, consecuentemente,  $Q=1$  en  $t=t_1+2\Delta t$ . Cuando el pulso de reloj vuelve a cero ( $Ck=0$ ) en  $t_2$ , *S* vuelve a 1 en  $t=t_2+\Delta t$  dejando  $Q=1$ . Cuando *D* pasa a cero en  $t_3$ , *B* pasa a 1 en  $t=t_3+\Delta t$  y *A* pasa a cero en  $t=t_3+2\Delta t$ .

Tras el flanco de subida del reloj en el tiempo  $t_4$ , *R* cambia a 0 en  $t=t_4+\Delta t$  y *Q* pasa también a cero un retardo más tarde, en  $t=t_4+2\Delta t$ . El cambio en el valor de *D* en  $t_5$  no se registra hasta el flanco de bajada de la señal de reloj, *Ck*, en  $t_6$  cuando *R* vuelve a 1 (en  $t_6+\Delta t$ ) y *B* pasa a 0 (en  $t_6+2\Delta t$ ) y  $A=1$  (en  $t_6+3\Delta t$ ). Así, en  $t_7+2\Delta t$ , *Q* se pone a 1 tras el flanco de subida del pulso de reloj en  $t_7$  y la bajada de *S* en  $t_7+\Delta t$ .



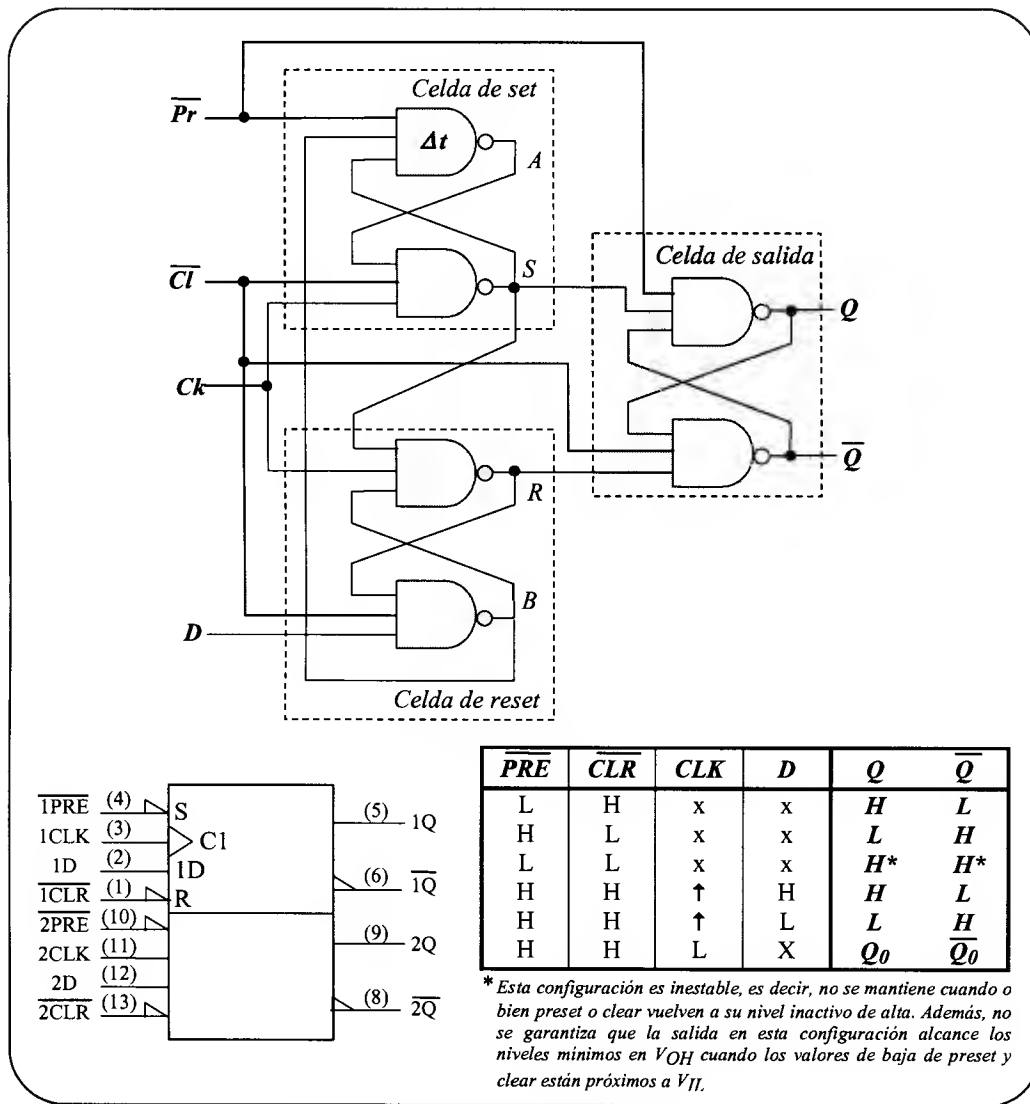
**Figura 8.28.** Descripción por retardos del funcionamiento del biestable D disparado por flancos y sintetizado con puertas NAND ( $S=R=0$  está prohibido).

Obsérvese que si hay nuevos cambios en  $D$  (por ejemplo el de  $t = t_8$ ), ya no se propaga hacia la salida hasta que se cumplan de nuevo las condiciones apropiadas, aunque el cambio si que es capturado por las celdas de entrada cuando  $B$  pasa de nuevo a 1 (en  $t = t_8 + \Delta t$ ) y  $A$  pase de nuevo a 0 (en  $t = t_9 + 2\Delta t$ ). Así se aísla, con una idea análoga a la de la configuración maestro-esclavo, la captura de la orden de su ejecución, definiendo un intervalo temporal, entre dos transiciones positivas de los pulsos del reloj, durante el cual el biestable es un elemento de memoria que almacena un bit en el estado de su celda de salida ( $Q$ ).

Si al circuito anterior (*figura 8.28*) le añadimos entradas de Preset ( $\overline{Pr}$ ) y Clear ( $\overline{Cl}$ ) y lo duplicamos, tenemos el circuito integrado SN74LS74A que contiene dos biestables D disparados por flancos positivos con entradas asíncronas de puesta a "1" ( $\overline{Pr} \overline{Cl} = 01$ ) y puesta a "0" ( $\overline{Pr} \overline{Cl} = 10$ ) activas en baja, para establecer condiciones iniciales. Para dejar el control a la entrada síncrona  $D$ , *preset* y *clear* permanecen en alta, tal como se muestra en la tabla de función de la *figura 8.29*, que incluye también el símbolo lógico.

Los circuitos biestables (D, T, R-S y J-K), y en particular el J-K, del que se pueden obtener todos los demás, cumplen en lógica secuencial funciones de síntesis modular análogas a las que cumplen las puertas NAND y NOR en lógica combinacional. Es decir, el biestable J-K es el módulo básico para el diseño en lógica secuencial. Cualquier autómata finito, y por consiguiente, cualquier función secuencial necesaria en un sistema de cálculo digital, puede ser sintetizada usando biestables J-K.

Los usos más frecuentes de estos circuitos binarios son la *síntesis de contadores, registros de desplazamiento, memorias RAM y macroceldas de salida en lógica secuencial programable*. Todas estas funciones serán estudiadas en temas posteriores y existen integradas en MSI, LSI y VLSI. Sin embargo, pueden existir aplicaciones que exijan el uso de biestables en pequeña escala. Por eso comentamos aquí algunos ejemplos de las soluciones integradas en SSI. Existen en prácticamente todas las familias lógicas (TTL, ECL, MOS y CMOS). A la hora de usar los catálogos es útil distinguir entre "latches" y "flip-flops". Ambos son biestables y la distinción en la nomenclatura procede del método de disparo. Cuando el disparo está sincronizado a niveles de la señal de reloj, se denomina "latch" y así aparecen los D. Cuando el disparo se produce únicamente durante las transiciones de la señal de reloj, se denomina "flip-flop", y se ofertan generalmente en J-K y en D.



**Figura 8.29.** Circuito SN74LS74A. Dos biestables D disparados por flancos positivos con *preset* y *clear*. (a) Diagrama lógico. (b) Tabla de función. (c) Símbolo lógico.

Así por ejemplo en TTL, tenemos circuitos tipo SN74ALS114A que incluye 2 biestables J-K disparados a flancos, soluciones duales para biestables D (AS874), soluciones múltiples con salidas de tres estados (ALS374) etc. Este último circuito consta de ocho biestables D con salidas de tres

estados y acceso para carga en paralelo. Los biestables se disparan en el flanco positivo del reloj. Hay una entrada para controlar la salida que permite colocar las ocho salidas en los estados lógicos normales (alta o baja) o bien un estado de alta impedancia, para operar en sistemas organizados en bus. De hecho el circuito es un impulsor bidireccional de bus. Mientras la señal de control está en alta impedancia, no afecta al funcionamiento interno de los biestables. Se pueden retener los últimos valores de D o entrar otros nuevos.

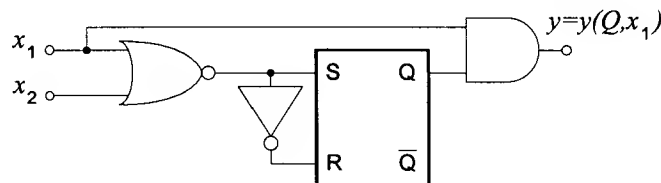
## 8.6. PROBLEMAS

En este tema hemos introducido primero la teoría de autómatas finitos como modelo matemático de la electrónica secuencial viendo que todo lo que nos hacía falta para su síntesis ya lo *conocíamos salvo los circuitos biestables. El resto del tema se ha dedicado al estudio de las distintas configuraciones de disparo (R-S, J-K, T y D).*

Por consiguiente, estos ejercicios se encaminan a repasar estos dos apartados (teoría de autómatas y biestables) en su doble vertiente: *análisis* y *síntesis*. En los ejercicios de análisis propondremos un circuito y preguntaremos por cuál es la forma de onda a la salida para una configuración determinada en las entradas. O bien, pediremos las funciones lógicas que las enlazan.

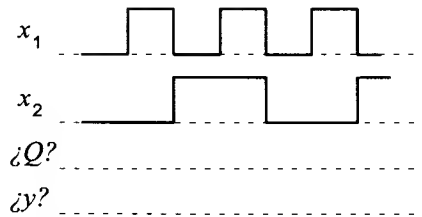
En síntesis propondremos la tarea inversa. Dada una función, ¿cómo puedo conseguir un circuito que la realice?. En algunos casos esta función será de carácter general. En otros corresponderá a situaciones ya conocidas en las que cambiamos NAND por NOR, por ejemplo, para obligar al alumno a que siga el mismo camino deductivo usado en el material teórico pero con otras puertas lógicas.

**E.8.1.** *Dibujar el diagrama de transición de estados, las tablas de transición y la expresión lógica de las funciones de producción de estados y de salidas ( $f$  y  $g$ ) para el circuito secuencial de la figura, suponiendo que inicialmente el biestable está en baja.*

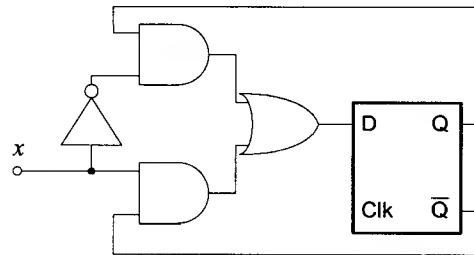




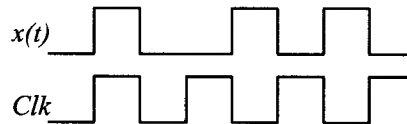
¿Cuál sería la secuencia de salida en  $y=y(Q,x_1)$ , si a la entrada aparece la siguiente secuencia?



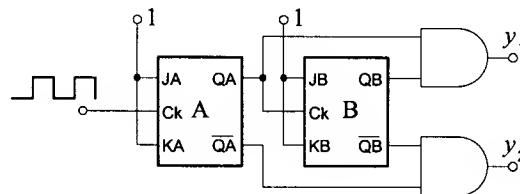
**E.8.2.** Dibujar el diagrama de transición de estados y producción de salidas, las tablas de transición y la expresión lógica de las funciones  $f$  y  $g$  para el circuito secuencial de la figura.



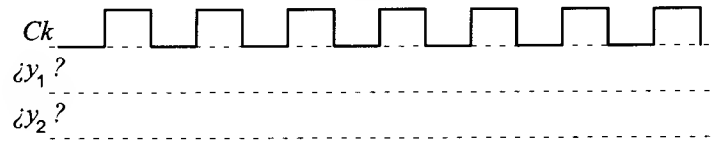
Obsérvese que ahora hay realimentación desde la salida y por consiguiente a la entrada se calculan funciones de  $x(t)$  y  $Q(t)$  pero  $Q(t)$  procede de los valores de  $D$  en  $(t-\Delta t)$ . Si el biestable  $D$  se dispara a subidas dibujar la evolución temporal de la señal en  $\bar{Q}$  cuando en la entrada  $x(t)$  y en el reloj  $Ck$ , aparecen las siguientes señales:



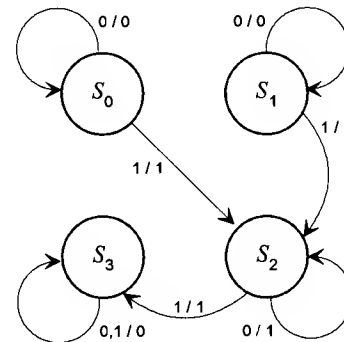
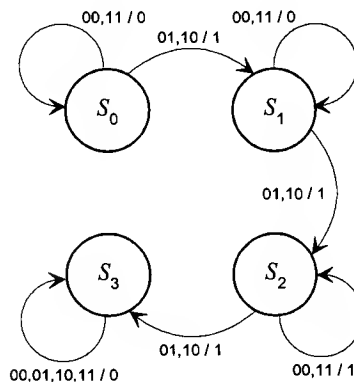
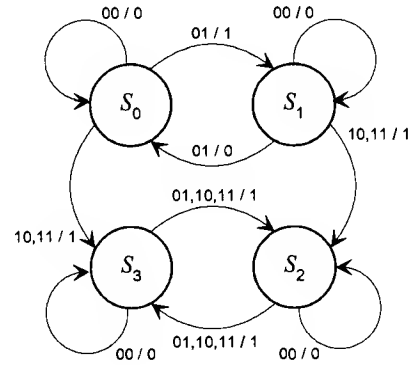
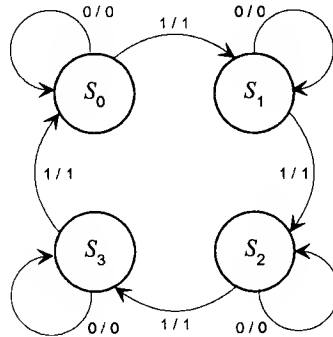
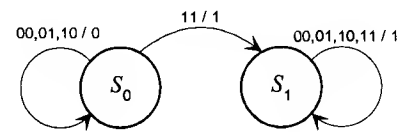
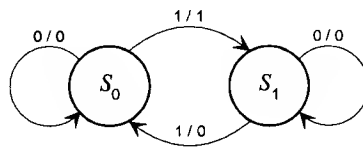
**E.8.3.** Analizar el circuito de la figura obteniendo el diagrama de transición de estados y la forma de onda en las salidas  $y_1$  y  $y_2$  durante siete pulsos sucesivos del reloj.



Obsérvese que al incluir dos biestables se trata de un autómata de cuatro estados y recuérdese también la forma de conseguir una configuración  $T$  usando J-K. La única entrada externa en este circuito es el reloj  $Ck$ .



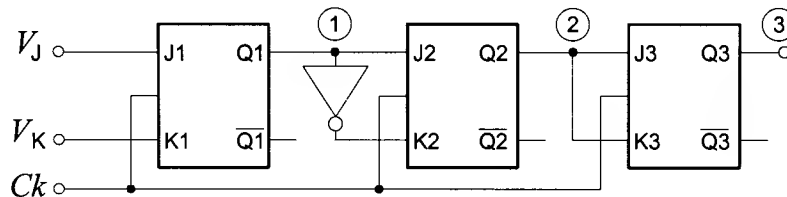
**E.8.4.** Sintetizar usando biestables  $D$  y las puertas lógicas necesarias los siguientes autómatas finitos de dos y cuatro estados.

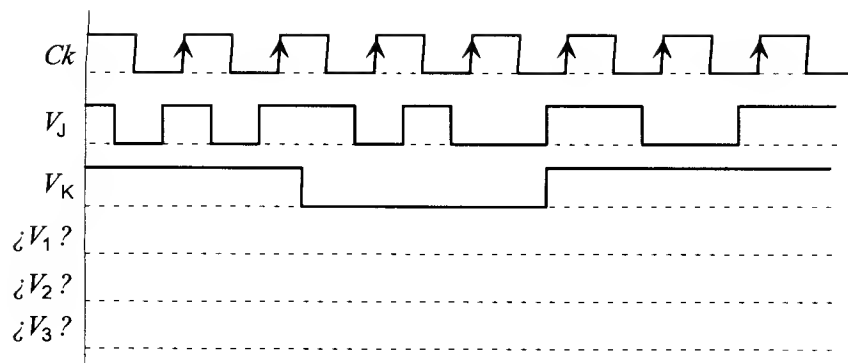


- E.8.5.** Diseñar un biestable R-S mínimo disparado a niveles usando sólo puertas NAND. Recordar que en una puerta NAND basta con una entrada en baja para que la salida esté en alta, sea cual fuere el valor de las otras entradas. Partiendo del diseño anterior introducir el reloj (Ck) para que el disparo sea síncrono. Por último modificar este circuito introduciendo las entradas adicionales de "preset" y "clear".
- E.8.6.** Obtener un biestable J-K primero a partir de un biestable tipo D y después a partir de uno de tipo T.
- E.8.7.** La figura adjunta muestra un circuito formado por la conexión serie de tres biestables J-K "master-slave" disparados por pulsos. El primero actúa como tal J-K, sin ninguna restricción en los valores de J y K. El segundo actúa como D y el tercero como T. Dibujar las formas de onda que se obtienen en los puntos (1), (2) y (3) de la figura cuando a la entrada del primer J-K se aplican las señales que se muestran en la figura.

Conviene recordar que el J-K "master-slave" utiliza los dos flancos del pulso de reloj para realizar sus cambios de estado. Durante la subida del pulso se produce el cambio de estado interno de la salida de la sección "master", que actúa de entrada a la sección "slave" y durante la bajada del pulso de reloj, se produce el cambio en la salida de la sección "slave" que es la que se observa externamente.

Esto significa que para construir el cronograma que se pide en este ejercicio hay que fijarse en los valores de J y K cuando suba el pulso de reloj para dibujar los valores en ( $Q$ ,  $\bar{Q}$ ) cuando el pulso baje.





# Preparación de la Evaluación

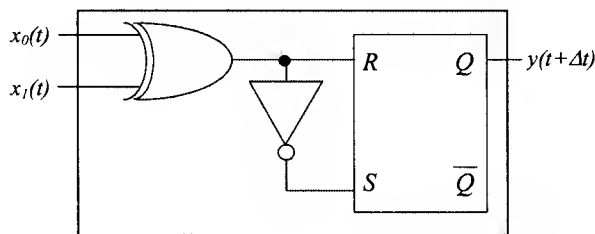
## Objetivos

- 1: Introducción a los Automatas Finitos. Concepto de Estado
- 2: Configuraciones Biestables: R-S, J-K, T y D.

## 1: Introducción a los Automatas Finitos. Concepto de Estado

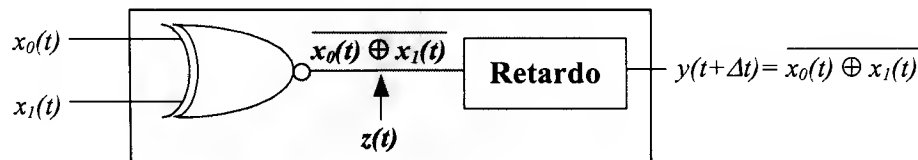
1.1. Buscamos ejemplos que nos ayuden a evaluar el grado de comprensión alcanzado sobre el concepto de estado. Para ello vamos a “descubrir” el procedimiento seguido en el texto para el caso del biestable “bolígrafo” con otro ejemplo sencillo.

a) Analice el comportamiento del circuito de la figura adjunta y trate de obtener la tabla de verdad que le adjuntamos.



$y(t)$	$x_0(t)$	$x_1(t)$	$R$	$S$	$y(t+\Delta t)$
0	0	0	0	1	1
1	0	0	0	1	1
0	0	1	1	0	0
1	0	1	1	0	0
0	1	0	1	0	0
1	1	0	1	0	0
0	1	1	0	1	1
1	1	1	0	1	1

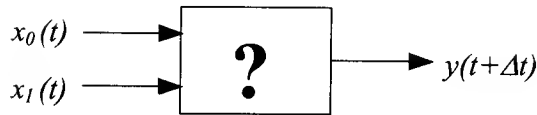
Evidentemente, no le costará llegar a la conclusión de que la expresión lógica es:



es decir,  $y(t + \Delta t) = z(t) \equiv S(t) = \overline{x_0(t)} \oplus x_1(t)$

b) Ahora que ya tiene las tablas y conoce el circuito y su función lógica suponga que no conoce el circuito y que sólo dispone de la información parcial que le da la medida de sus

entradas y salidas de acuerdo con esta nueva tabla:



$y(t)$	$x_0(t)$	$x_1(t)$	$y(t+\Delta t)$
0	0	0	1
1	0	0	1
0	0	1	0
1	0	1	0
0	1	0	0
1	1	0	0
0	1	1	1
1	1	1	1

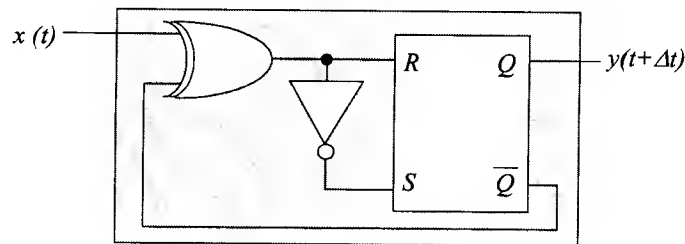
**b.1.** Trate de “descubrir” la nueva variable auxiliar,  $z(t)$ , necesaria para que la tabla corresponda a un circuito combinacional real.

**b.2.** Reflexione sobre el nacimiento de la secuencialidad al asociar la variable auxiliar,  $z(t)$ , a la entrada del biestable D, de forma que la salida,  $y(t+\Delta t)$ , es la entrada al biestable en el instante anterior.

**1.2.** Repita el ejercicio anterior con la modificación introducida ahora en el circuito (una realimentación interna de  $\bar{Q}$  a la puerta NOR).



**b)** “Oculté” el circuito y compruebe la necesidad de introducir la variable  $z$  y el tiempo.

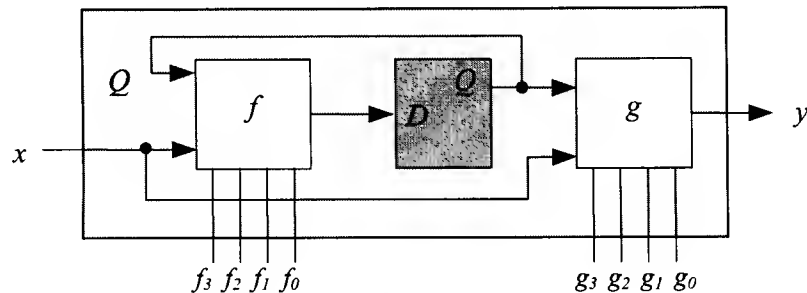


**1.3.** Revise el ejercicio E.8.1 del texto de problemas y repita el análisis cambiando el biestable R-S que opera como D por un J-K operando como T.

**1.4.** Revise los ejemplos de síntesis (casos a y b) que se resuelven en el ejercicio E.8.4 del texto de problemas. Como se ve allí, se trata de pasar del diagrama de transición de estados a las expresiones lógicas de la variable de salida y de las entradas de disparo de los biestables.

Diseñe ahora un circuito general para autómatas de dos estados, una variable de entrada,  $x_0$ , y una variable de salida,  $y$ , de forma que sea cual fuere el diagrama de transición de estados que nos propongan podamos sintetizarlo programando este autómata general. Es decir, poniendo a 1 ó a 0 los bits que controlan las funciones de producción del nuevo estado ( $f$ ) y de la producción de la salida a partir de la entrada y el estado ( $g$ ). Aconsejamos pensar en un esquema como el de la figura adjunta en el que ( $f_0, f_1, f_2, f_3$ ) son los bits de control de los términos mínimos de  $f$  y ( $g_0, g_1, g_2, g_3$ ) los bits que controlan  $g$ .

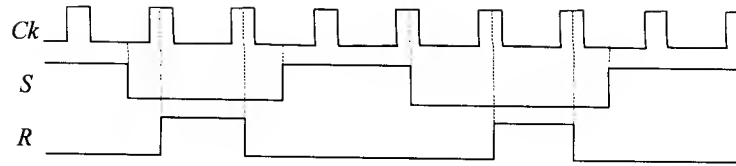
Usar los conocimientos del tema 7 sobre PROM, PAL y PLA para decidir la forma más conveniente de sintetizar las funciones universales de 2 variables.



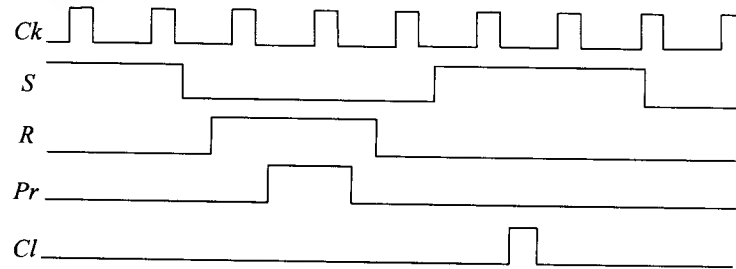
- 1.5. Revise ahora la segunda parte del ejercicio E.8.4 del texto de problemas (casos c, d, e y f). Se trata de sintetizar, usando biestables D y puertas lógicas, varios casos de autómatas finitos de 4 estados con dos variables de entrada y una variable de salida.

## 2: Configuraciones Biestables: R-S, J-K, T y D.

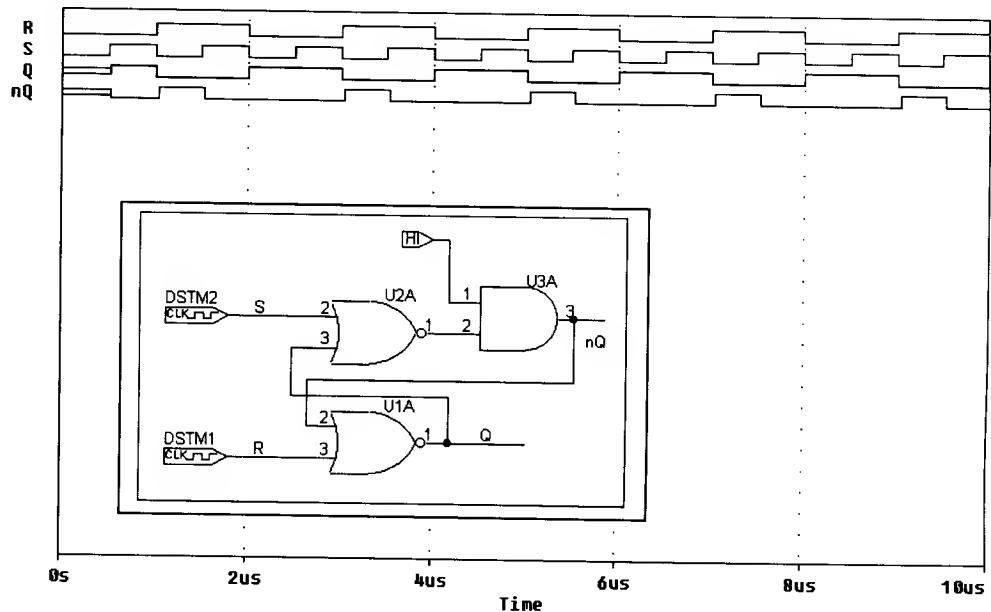
- 2.1. En la figura 8.12 hemos realizado un estudio detallado del comportamiento del latch R-S sintetizado con puertas NOR y en el problema E.8.5. del texto de problemas hemos visto la síntesis con puertas NAND. Repita para este caso la evolución de las realimentaciones en los ocho casos posibles (R, S y Q inicial iguales a : 000, 001, ..., 111).
- 2.2. ¿Por qué una celda R-S básica no es válida como elemento de memoria?. ¿Qué entradas adicionales necesita?.
- 2.3. ¿Cuántas formas de disparo conoce para los biestables R-S?. Obtenga la evolución de la salida Q para cada uno de estos tipos de disparo cuando las señales en R, S y Ck (reloj) son las de la figura?.



- 2.4. ¿Cómo se modificaría la evolución temporal de la salida de un biestable R-S disparado por flancos positivos al incluir las señales de preset y clear que se muestran en la figura adjunta?. (Suponga  $Q$  inicialmente en baja).



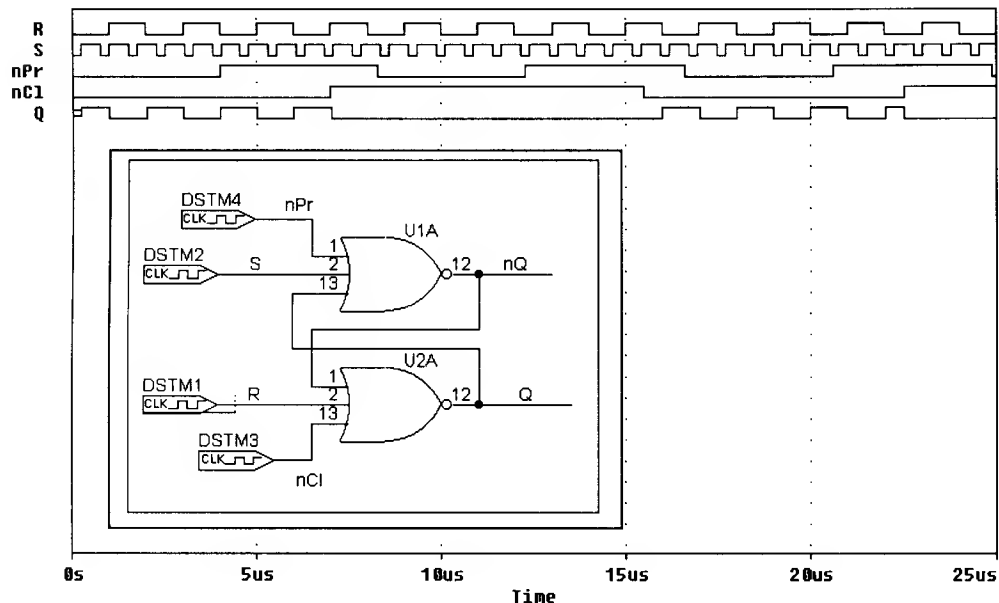
- 2.5. Simule el circuito de la figura adjunta y compruebe que el cronograma es el que se incluye en esta misma figura.



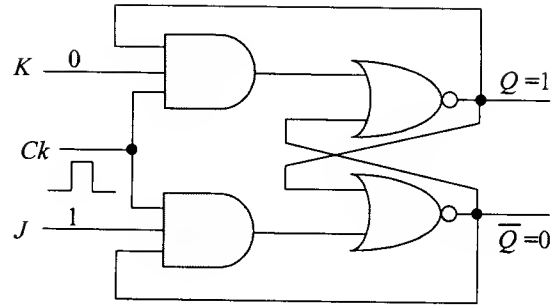


Le recordamos que en la figura 8.13 vimos el resultado de la simulación de la celda R-S básica y su cronograma, incluyendo la oscilación en la bajada simultánea de R y S. Allí comentamos que una forma de eliminar la oscilación era introducir un pequeño retardo en una de las ramas. Intente comprobar ahora esta situación con el circuito propuesto en este ejercicio.

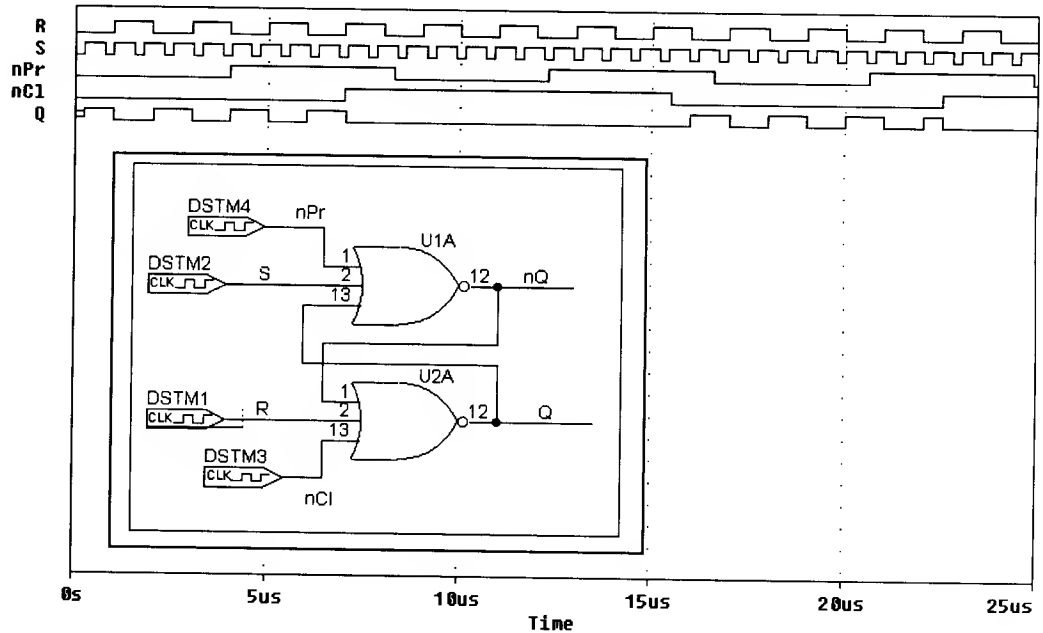
- 2.6. Compruebe el comportamiento asíncrono de los terminales de preset y clear usando el simulador. Como orientación le sugerimos que simule la configuración adjunta.



- 2.7. En la figura 8.20 del tema se muestra el análisis detallado del comportamiento de un biestable J-K sincronizado a niveles para la configuración conflictiva  $J=K=1$  con el estado inicial  $Q=0$ . Intente ahora reconstruir este proceso de análisis para otras configuraciones más sencillas. Por ejemplo para  $(J=1, K=0 \text{ y } Q=1)$  y para  $J=0, K=1 \text{ y } Q=0$ . Recuerde el procedimiento consiste en considerar que cada puerta, AND ó NOR, independientemente de su función lógica, introduce un retardo,  $\Delta t$ , de forma que para entradas presentes en un cierto  $t$  los resultados del cálculo realizado por la puerta sólo los tendrá accesibles en  $t+\Delta t$ . Y estos resultados son a su vez variables de entradas para la puerta que viene detrás siguiendo el lazo de realimentación.



2.8. Comprobar, usando un simulador tipo PSpice, el funcionamiento de un biestable J-K con Preset y Clear. Por ejemplo, para el circuito 74109 se obtendría el diagrama de tiempos de la figura. Explicar su funcionamiento y obtener la tabla de verdad a partir del diagrama de tiempos.



El contenido de esta tema es muy adecuado para el uso de un simulador. Las funciones secuenciales asociadas a contadores y registros de desplazamiento, así como el comportamiento de los propios biestables T, D, J-K y R-S, están estructuradas conceptualmente sobre la evolución temporal de un conjunto de señales. Es decir, sobre un cronograma. Así, es difícil comprender el funcionamiento de estos circuitos si no somos capaces de seguir la evolución temporal de las variables de entrada y de estado. De forma complementaria, cuando se comprende un cronograma hemos comprendido, de hecho, el funcionamiento de un circuito secuencial. Por eso consideramos que es una buena estrategia docente el hacer énfasis en el uso de un simulador para la construcción de los cronogramas asociados a biestables, contadores y registros.

## REFERENCIAS BIBLIOGRÁFICAS

1. Gajski, D.D.: *“Principios de Diseño Digital”* Prentice Hall, 1998.
2. Sandige, R.S.: *“Modern Digital Design”*. McGraw-Hill. 1990.
3. Texas Instruments. *“The TTL Data Book”*. 1987.
4. [www.fairchildsemi.com](http://www.fairchildsemi.com)



# TEMA 9

---

## ***Introducción al Diseño Secuencial: Contadores y Registros***

---

### **+ CONTEXTO**

*En el tema anterior hemos visto una introducción a la teoría modular de autómatas finitos y determinísticos que es el modelo matemático de la electrónica secuencial, de forma análoga a como el álgebra de Boole era el soporte de la electrónica combinacional.*

*Vimos allí que para la síntesis secuencial necesitábamos funciones de lógica combinacional que ya habíamos estudiado en temas anteriores y retardos (biestables D) para almacenar el estado del sistema. A continuación, estudiamos los distintos tipos de biestables y la forma de sintetizarlos a partir de la configuración más general que era la del J-K.*

*Ahora vamos a presentar el diseño secuencial con cierta generalidad y a ilustrarlo con funciones específicas (contadores y registros) usadas frecuentemente en todos los sistemas digitales. Queremos dejar claro sin embargo que el procedimiento de síntesis de circuitos secuenciales es completamente general.*

*Lo peculiar de la presentación que se ofrece aquí de la teoría de autómatas en su orientación hacia la lógica secuencia programable (PLDs avanzadas y FPGAs) de forma que cuando dibujamos el esquema del circuito secuencial (parte combinacional, banco de celdas de memoria, realimentación, ...) el modelo matemático usado tiene una muy directa y obvia relación con la arquitectura de las CPLDs: primero la parte combinacional programable, después los biestables D y, finalmente, la realimentación y, potencialmente, la parte combinacional de otra CPLD para la producción de salidas a partir de la entrada y el estado del autómata.*

## **++ CONOCIMIENTO PREVIO NECESARIO**

*Ya tenemos todo lo que necesitamos: un nuevo modelo matemático y los elementos de memoria. Con esto y el contenido de los temas sobre lógica combinacional programable y no programable ya podemos abordar el diseño secuencial en general y, en particular, el de las dos funciones intermedias básicas: los contadores y los registros de desplazamiento.*

## **+++ OBJETIVOS DEL TEMA**

- Objetivo 1:** *Saber sintetizar circuitos secuenciales. Es decir, saber obtener un circuito a partir de la descripción en lenguaje natural de la función que queremos que realice. Además, debemos saber llegar hasta el final en el proceso de síntesis usando biestables D ó J-K.*
- Objetivo 2:** *Conocer la estructura interna (puertas y biestables) y el comportamiento externo de los distintos tipos de circuitos contadores. Saber manejar los datos de catálogo correspondientes y tener ciertos conocimientos sobre la función de estos circuitos cuando se integran en arquitecturas más complejas.*
- Objetivo 3:** *Conocer la estructura interna y el comportamiento de los distintos tipos de registros de desplazamiento. Saber manejar los datos de catálogo correspondientes y tener ciertos conocimientos sobre la función de estos circuitos cuando se integran en arquitecturas más complejas.*

## ★★★★ GUÍA DE ESTUDIO

*Para fijar ideas sobre la línea metodológica seguida en el desarrollo de este material didáctico, conviene recordar que en este tema dedicado al diseño secuencial estamos en la misma posición relativa en la que nos encontrábamos en lógica combinacional al comienzo del tema 5. Es decir, tenemos el modelo matemático y la síntesis de un conjunto completo de operadores (por ejemplo, puertas NAND y retardos D) y debemos contestar a preguntas equivalentes a las que nos hicimos en el tema 5°:*

- a) *¿Cuáles son las funciones secuenciales de nivel intermedio que hacen falta para la síntesis de los grandes sistemas digitales de instrumentación, cálculo, comunicación y control?.*
- b) *¿Cómo podemos sintetizar esas funciones intermedias usando biestables (D, T, R-S, J-K) y las funciones de lógica combinacional que ya conocemos?.*

*La contestación en extenso a la primera pregunta es complicada y queda fuera del alcance de este curso. Entre otras razones porque sería necesario conocer con precisión las necesidades computacionales de todas y cada una de estas aplicaciones antes de pasar a su síntesis. Es evidente sin embargo que sea cual fuese esa necesidad, si es computable, siempre existe un autómatas finito que la describe y este autómatas es sintetizable, por ejemplo, mediante PLDs.*

*Si nos limitamos dentro de los sistemas digitales a los sistemas de cálculo y buscamos sólo las necesidades básicas y de uso más frecuente en arquitectura de ordenadores, ya podemos afirmar que las funciones secuenciales intermedias son:*

- a.1. *Contadores*
- a.2. *Registros de desplazamiento*
- a.3. *Temporizadores (monoestables) y relojes (astables, relojes polifásicos, etc...).*
- a.4. *Memorias RAM (estáticas y dinámicas, con organización FIFO, LIFO, CAM, etc...).*

*La contestación a la segunda pregunta (¿cómo podemos sintetizarlas?) es el objetivo del resto de los temas. Es decir, ahora buscamos la especificación, análisis y síntesis de este conjunto mínimo de funciones secuenciales intermedias (contadores, registros, relojes y memorias) esenciales para comprender la estructura interna de un ordenador.*

*Dos reflexiones pueden ayudarnos en el estudio del tema. La primera es de procedimiento. Como siempre, usaremos circuitos sencillos, con pocos bits, para explicar las funciones de “contar” y “desplazar”. Veremos después cual es el modelo formal subyacente y, finalmente, daremos “pistas” para continuar el estudio con circuitos reales de mayor complejidad y para conectar con el uso que se hace de estos circuitos en otras asignaturas.*

*La segunda reflexión ya la hemos mencionado en otros temas, y se refiere a la necesidad de estudiar de forma coordinada los contenidos de la Electrónica Digital. Es importante, a medida que avanzamos en el estudio de las funciones de la electrónica secuencial, que el alumno recuerde que la Electrónica Digital es el soporte físico de la Computación pero las funciones de “contador”, “registro”, “memorias RAM”, “memorias pila” (FIFO), “memorias asociativas”, etc... se repiten en el estudio de la Arquitectura de Computadores y se vuelven a repetir en el nivel software. Esta visión global creemos que puede ayudar al alumno a economizar esfuerzos y reutilizar ventajosamente sus conocimientos. Si sabe qué es una memoria CAM en Electrónica va a aumentar su comprensión de los procesos asociativos en la arquitectura de las memorias caché y a nivel de programación, por ejemplo.*



---

## ***Contenido del Tema***

---

### **9.1. INTRODUCCIÓN AL DISEÑO SECUENCIAL CON BIESTABLES D, T Y J-K.**

Los sistemas secuenciales pueden ser síncronos (guiados por los pulsos de un reloj) o asíncronos (guiados por cambios de nivel en las variables; sin que tengan que coincidir con los pulsos del reloj). La síntesis síncrona es más sencilla y sólo de ella hablaremos a nivel general. Sin embargo, comentaremos la síntesis asíncrona para alguna función específica de baja complejidad (contadores).

La síntesis de un circuito secuencial síncrono se puede hacer usando cualquier tipo de biestable. La forma más natural, aunque no da lugar a realizaciones mínimas, es la D, ya que en este caso la transcripción de las funciones de transición es inmediata. Es decir, como la ecuación del biestable D es simplemente,  $Q_{n+1} = D$ , es trivial el obtener la función de excitación ( $D$ ) necesaria para que el estado ( $Q_{n+1}$ ) sufra una determinada transición (paso de  $Q_n$  a  $Q_{n+1}$ ). El estado final al que queremos que llegue el biestable D debe de coincidir con la función de excitación, tal como se ilustra en la tabla de la *figura 9.1*.

En un proceso de análisis, los datos son las entradas ( $D$  y el *reloj*) y la función que queremos obtener es la dependencia de la salida, ( $Q_{n+1}$ ), con el valor de la entrada y del estado anterior, que en los D es inmediato:

$$Q_{n+1} = D \quad [9.1]$$

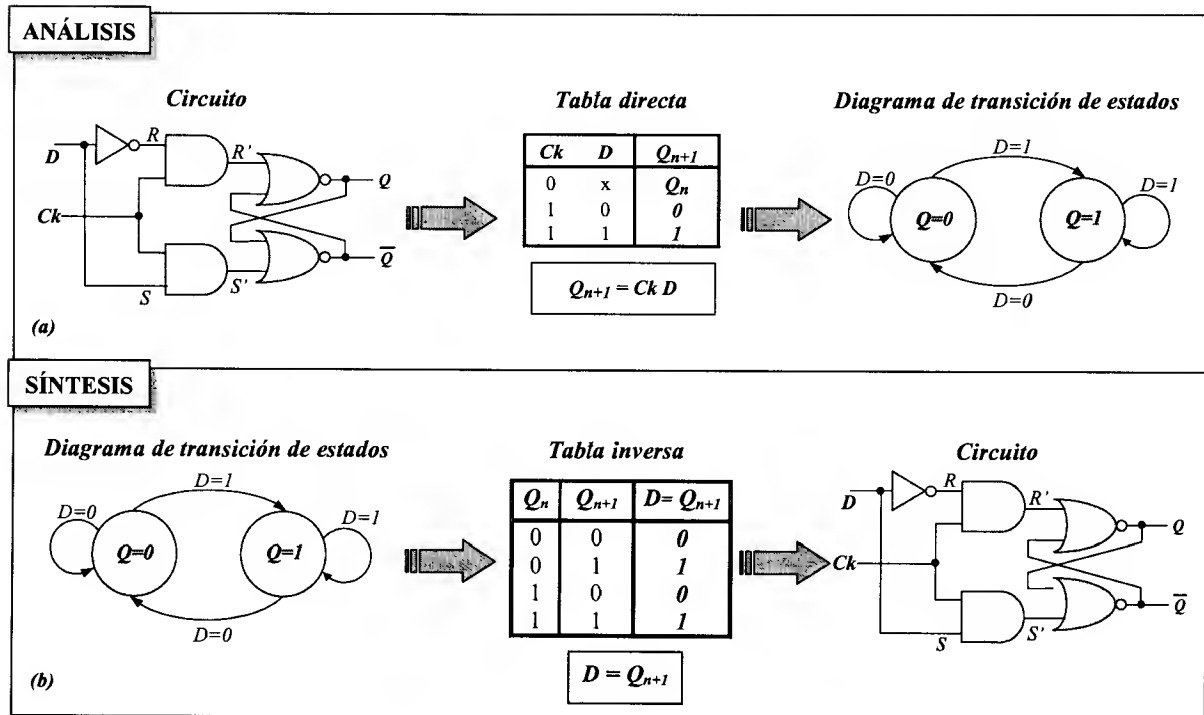
En cambio, cuando tenemos que usar un conjunto de biestables D en la síntesis de un circuito secuencial concreto, el proceso es el inverso. Ahora el dato, lo que sabemos, es el diagrama de transición de estados que necesitamos sintetizar. Es decir, las relaciones entre  $Q_n$  y  $Q_{n+1}$  para todos y cada uno de los biestables que forman el estado. En el caso del biestable D estas relaciones son las que se muestran en la parte (*b*) de la *figura 9.1* y son también elementales:



$$D = Q_{n+1}$$

[9.2]

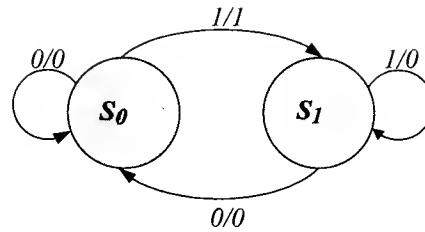
Es decir, hay que poner en  $D$  lo que se quiera tener a la salida ( $Q_{n+1}$ ) en el próximo pulso de reloj.



**Figura 9.1.** Ilustración sobre un biestable D de los procesos de análisis y síntesis. **(a)** Análisis : Dado  $D$ , calcular  $Q$ . **(b)** Síntesis: Dado el  $Q_{n+1}$  que necesito, ¿qué  $D$  debo usar?. Es decir, ¿cuál es la función de excitación, a partir de la tabla de transición de estados.?

Una vez conocida la función de excitación del biestable D, ya podemos usarla en la síntesis de circuitos más complejos con  $2^N$  estados que usan  $N$  biestables D para su síntesis.

**Ejercicio:** Sintetizar, usando un biestable D, el circuito secuencial correspondiente al autómata de dos estados de la figura.



**Solución:**

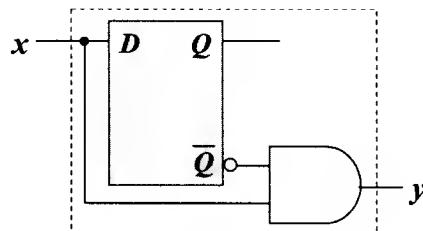
Como tiene dos estados, asociamos  $S_0$  a  $Q = 0$  y  $S_1$  a  $Q = 1$ . Llamamos  $x$  a la entrada e  $y$  a la salida y obtenemos las tablas de verdad volviendo a escribir la información contenida en el diagrama de transición de estados teniendo en cuenta que las transiciones están marcadas por  $x/y$ , que en el caso más general es: *configuración de entrada/configuración de salida*

$x$	$Q_n$	$Q_{n+1}$	$y$	$D$
0	0	0	0	0
0	1	0	0	0
1	0	1	1	1
1	1	1	0	1

**Figura 9.2.** Tabla de verdad

Así, de la observación de la primera y la última columna de la tabla vemos que  $D=x$ . Evidentemente, en un caso más complicado tendríamos que minimizar la expresión  $D=f(x, Q_n)$ .

Análogamente,  $y = x \overline{Q}_n$  y por consiguiente el circuito secuencial del ejemplo es:



**Figura 9.3.** Circuito solución

Veamos ahora la tabla de excitación para el biestable T repitiendo, en la *figura 9.4*, el contenido

equivalente al de la *figura 9.1*, pero cambiando las tablas directas e inversas por las correspondientes al T. Obsérvese que lo único que hemos hecho ha sido volver a escribir las tablas de verdad. Ya sabíamos desde el tema anterior que "cuando  $T=0$  el biestable T no cambia de estado y cuando  $T=1$ , el biestable cambia de estado"

Es decir, 
$$Q_{n+1} = T \cdot \overline{Q}_{n+1} + \overline{T} \cdot Q_n \quad [9.3]$$

Ahora, en *síntesis*, planteamos la pregunta de forma inversa: *¿qué valor necesito poner en T para que el biestable realice cada una de las cuatro transiciones posibles ( $Q_n \rightarrow Q_{n+1}$ )?*

Este valor es el que obtenemos de la expresión

$$T = \overline{Q}_n \cdot Q_{n+1} + Q_n \cdot \overline{Q}_{n+1} \quad [9.4]$$

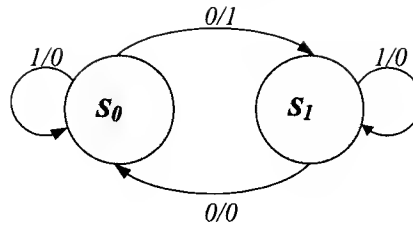
que a su vez ha sido obtenida por simple reescritura de la tabla de análisis considerando ahora a  $Q_{n+1}$  y  $Q_n$  como dato de entrada y obteniendo  $T$  como función de estas dos variables.

Análisis			Síntesis		
$T$	$Q_n$	$Q_{n+1}$	$Q_n$	$Q_{n+1}$	$T$
0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	1	1	0
$Q_{n+1} = T \overline{Q}_n + \overline{T} Q_n$			$T = \overline{Q}_n Q_{n+1} + Q_n \overline{Q}_{n+1}$		

**Figura 9.4.** Obtención de la función de excitación en el biestable T a partir de su tabla de transición de estados.

---

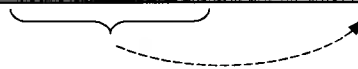
**Ejercicio:** *Diseñar, usando un biestable T, un circuito secuencial cuyo diagrama de transición de estados sea:*



**Solución:**

Asociamos  $S_0$  a  $Q=0$  y  $S_1$  a  $Q=1$ . Llamamos  $x$  a la entrada e  $y$  a la salida y pasamos la información del diagrama de transición de estados a la tabla de verdad.

$x$	$Q_n$	$Q_{n+1}$	$y$	$T$
0	0	1	1	1
0	1	0	0	1
1	0	0	0	0
1	1	1	0	0

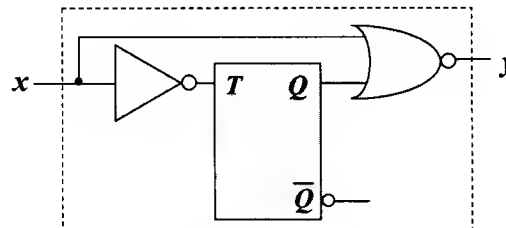


**Figura 9.5.** Tabla de verdad

$$Q_{n+1} = \bar{x} \cdot \bar{Q}_n + x \cdot Q_n \quad [9.5]$$

$$y = \bar{x} \cdot \bar{Q}_n = \overline{x + Q_n}, \quad x = \bar{Q}_n \cdot \bar{Q}_{n+1} + Q_n \cdot Q_{n+1} \quad [9.6]$$

Como sabemos que  $T = \bar{Q}_n \cdot Q_{n+1} + Q_n \cdot \bar{Q}_{n+1}$ , obtenemos que la entrada del biestable,  $T$ , debe ser la negación de la entrada  $x$ . Así,  $T = \bar{x}$ , y el circuito:



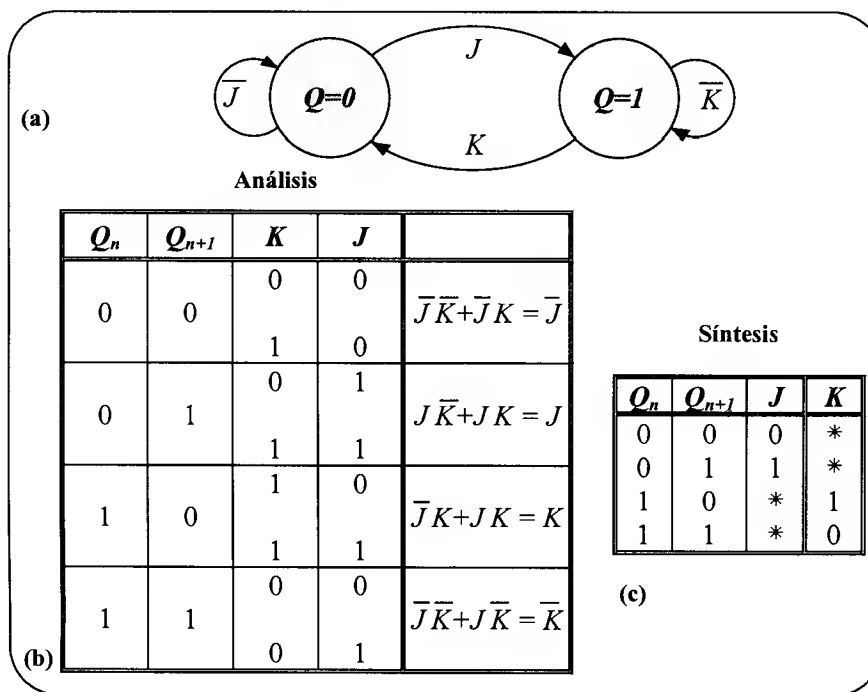
**Figura 9.6.** Circuito solución

Veamos finalmente el caso de los biestables J-K. Si quisiéramos realizar la síntesis de estas transiciones de estado ( $Q_n \rightarrow Q_{n+1}$ ) con biestables J-K tendríamos que contestar a la siguiente pregunta:

*¿Cómo tienen que ser las funciones de excitación de cada uno de los biestables ( $J_1, K_1$ ;  $J_2, K_2$  y  $J_3, K_3$ , etc.) para que efectivamente provoquen las transiciones que marca la tabla de transición de estados?*

Para contestar a esta pregunta repetimos en la *figura 9.7* el esquema seguido en 9.1 y 9.4 y seguimos las tablas de las *figuras 9.7(a), (b)* y *(c)*. En 9.7(a) hemos repetido por comodidad el diagrama de transición de estados y en 9.7(b) la tabla que define el comportamiento del biestable J-K cuya expresión lógica es:

$$Q_{n+1} = J \cdot \bar{Q}_n + \bar{K} \cdot Q_n \quad [9.7]$$



**Figura 9.7.** Funciones de excitación en biestables J-K necesarias para conseguir cada una de las cuatro transiciones posibles en el estado. (a) Diagrama de transición de estados. (b) Tabla del J-K. (c) Tabla de control del cambio de estado obtenida a partir de la (b) (tabla de excitación).

Al igual que en los casos D y T, en 9.7(c) hemos construido la tabla inversa, es decir hemos escrito las cuatro configuraciones posibles de un estado,  $Q_n$ , y de su sucesor,  $Q_{n+1}$ , y hemos anotado a la derecha los valores que deberían tener  $J$  y  $K$  para provocar esas transiciones. Así, para pasar de  $Q_n=0$  a  $Q_{n+1}=1$ , por ejemplo, las entradas pueden ser  $J=1$ ,  $K=0$ , ó bien  $J=1$ ,  $K=1$ , lo que significa que el valor de  $K$  es irrelevante y la transición la marca  $J=1$ .

$$J\bar{K} + JK = J(\bar{K} + K) = J \quad [9.8]$$

De la misma forma razonaríamos para el resto de las configuraciones hasta obtener toda la tabla. Las posiciones que se cubren con una “\*” significan que es irrelevante el valor que toma esa variable. Es decir, la  $J$  manda las transiciones desde  $Q_n = 0$  (independientemente del valor de  $K$ ) y la  $K$  manda las transiciones desde  $Q_n = 1$  (independientemente del valor de  $J$ ).


**Ejercicio:** *Comprobar que se ha comprendido la síntesis con J-K, repitiendo ahora los ejercicios que hicimos para los biestables D y T, es decir, sintetizar usando un J-K los autómatas que antes sintetizamos con D y T.*

**Solución:**

Calculemos los valores de  $J$  y  $K$  a partir de las tablas de verdad.

- (a) Veamos el caso del primer ejercicio. Partimos de la tabla de verdad a la que hemos de añadirle dos nuevas columnas, una para  $J$  y otra para  $K$ . Estas columnas las rellenamos con los valores que deben tomar  $J$  y  $K$  respectivamente para que se cumpla las transiciones correspondientes de  $Q_n$  a  $Q_{n+1}$ . Así, la tabla es:

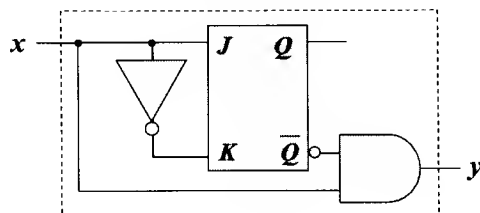
$x$	$Q_n$	$Q_{n+1}$	$y$	$J$	$K$
0	0	0	0	0	*
0	1	0	0	*	1
1	0	1	1	1	*
1	1	1	0	*	0



**Figura 9.8.** Tabla de verdad

Luego:  $J = x$  y  $K = \bar{x}$  [9.9]

y por supuesto, la salida sigue siendo  $y = x \bar{Q}_n$  y el circuito queda:



**Figura 9.9.** Circuito solución

Obsérvese que, al final, al poner el inversor entre  $x$  y  $K$ , lo que hemos hecho es emular un D con un J-K.

(b) Veamos ahora el caso del 2º ejercicio.

$x$	$Q_n$	$Q_{n+1}$	$y$	$J$	$K$
0	0	1	1	1	*
0	1	0	0	*	1
1	0	0	0	0	*
1	1	1	0	*	0

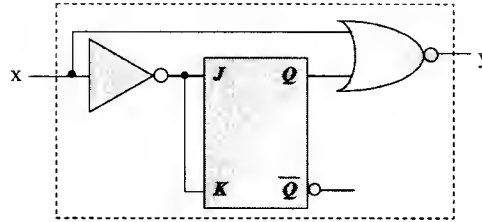
**Figura 9.10.** Tabla de verdad

Primero sacamos la  $J$  y la  $K$  a partir de las transiciones  $Q_n \rightarrow Q_{n+1}$ .

Después obtenemos esos valores necesarios en  $J$  y  $K$  a partir de  $x$  y de  $Q_n$  (en general)

$$J = \bar{x} \text{ y } K = \bar{x} \quad [9.10]$$

Es decir,  $J = K = \bar{x}$ , que significa que hemos emulado el biestable T usando un J-K. La salida no cambia,  $y = \overline{x + Q_n}$ , y el circuito resultante es:



**Figura 9.11.** Circuito solución

Hemos visto las funciones de excitación necesarias en los biestables D, T y J-K para producir cada una de las cuatro transiciones posibles en un elemento de memoria, en un bit. Sin embargo, en un diseño real trabajamos con registros de memoria de  $N$  bits. Es decir, con autómatas finitos de  $2^N$  estados sintetizados usando  $N$  biestables. Por consiguiente, para sintetizar el autómata necesitamos  $N$  conjuntos de ecuaciones de excitación, una por cada biestable. Así, para que un autómata de 8 estados ( $Q_2Q_1Q_0=000, 001, 010, \dots, 111$ ) pase del estado  $S_i=010$  al estado  $S_j=011$  el biestable del bit más significativo ( $Q_2$ ) no debe cambiar de estado ( $0 \rightarrow 0$ ), el segundo ( $Q_1$ ) tampoco ( $1 \rightarrow 1$ ) y el bit menos significativo ( $Q_0$ ) debe pasar de cero a uno ( $0 \rightarrow 1$ ).

Si quisiéramos sintetizar este autómata con tres biestables D ( $D_2, D_1, D_0$ ), las funciones de excitación necesarias para esa transición serían:

$$D_2=0, D_1=1, D_0=1 \quad [9.11]$$

Si por el contrario quisiéramos usar biestables J-K, las funciones de excitación de los tres biestables ( $J_2K_2, J_1K_1, J_0K_0$ ) tendrían que ser:

$$J_2=0, J_1=*, J_0=1 \quad [9.12]$$

$$K_2=*, K_1=1, K_0=* \quad [9.13]$$

## 9.2. PROCEDIMIENTO GENERAL DE SÍNTESIS

Vamos ahora a plantear el procedimiento general de síntesis secuencial síncrona de autómatas arbitrarios de los que los contadores y los registros sólo son casos particulares. Este procedimiento debe de incluir los siguientes pasos:



- P.1.** Descripción en *lenguaje natural* de forma clara, completa, precisa e inequívoca de la función que queremos sintetizar.
- P.2.** *Representación* de esa descripción en términos de *autómatas finitos*, especificando los espacios de entradas y salidas, el espacio de estados internos necesarios y las funciones de transición de estados y producción de salidas a partir de las entradas y los estados.
- P.3.** *Minimización* (en su caso) del número de estados. En un sistema secuencial hay tantos estados distintos como historias de estímulos distinguibles. Y no hacen falta más para duplicar la función. Si existen estados redundantes, conviene eliminarlos para conseguir una síntesis mínima sobre clases de equivalencia de estados.
- P.4.** *Selección de biestables* (D, T, J-K) y cálculo de las funciones de excitación correspondientes.
- P.5.** *Asignación de estados*. La asignación de estados es el procedimiento mediante el cual se hacen corresponder biestables específicos a cada uno de los bits resultado de la codificación en binario de los estados del autómata. Por ejemplo, podemos usar el siguiente procedimiento.
- P.5.1.** Sea  $A$  el número de estados del autómata. Para su síntesis necesitamos un *número de biestables*,  $N$ , tal que  $2^N \geq A \geq 2^{N-1}$ .
- P.5.2.** Entonces, *ordenamos* de forma arbitraria esos *estados* desde  $S_0=0$  hasta  $S_{A-1}=2^N-1$  y *rebeschimos en binario el subíndice* que identifica al estado:
- $$S_0 = (00\dots 0), S_1 = (00\dots 1), \dots, S_{A-1} = (11\dots 1)$$
- P.5.3.** Finalmente *elegimos un biestable*,  $(Q_0, Q_1, \dots, Q_{N-1})$ , para cada bit del estado.
- P.5.4.** *Aplicamos los algoritmos de síntesis* para obtener las funciones  $y_k(t) = D_k(t)$  de excitación de los  $N$  biestables  $D$ , a partir de las matrices de transición del autómata.
- P.6.** *Obtenemos de las funciones de excitación*, correspondientes a cada uno de esos  $N$  biestables.

Ya tenemos la *descripción formal* (P.2), la *minimización* (P.3), la *selección del tipo de biestable* con el que queremos realizar la síntesis (P.4) y la *asignación* de los bits que definen el estado (P.5). La última etapa del diseño es la *obtención de las funciones de excitación* ( $D_i$ ,  $T_i$  ó  $J_i-K_i$ ), de los  $N$  biestables,  $(Q_0, Q_1, \dots, Q_{N-1})$  que definen el estado del autómata.

### 9.3. REPRESENTACIÓN, SÍNTESIS Y ANÁLISIS MODULAR DE AUTÓMATAS CON PLDS

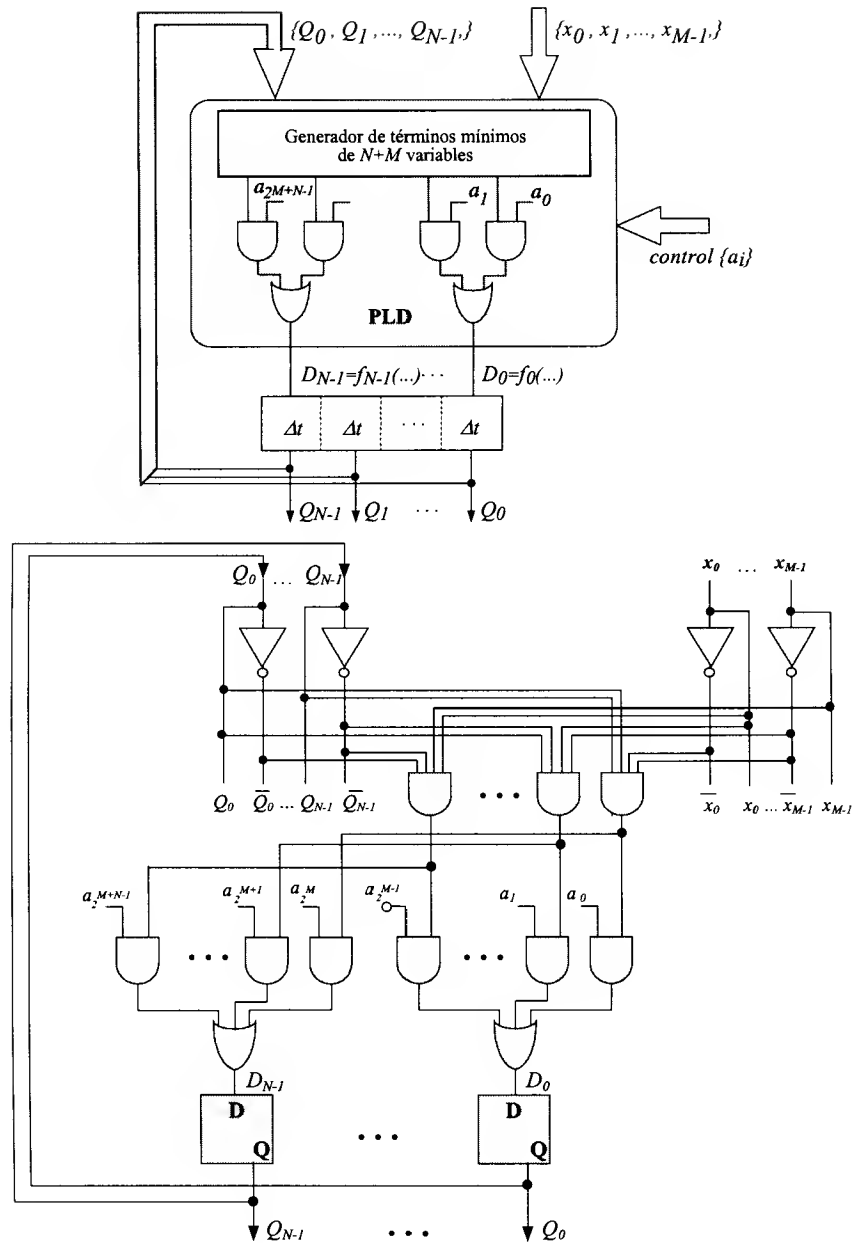
En la figura 8.7 del tema anterior vimos la arquitectura general de un circuito secuencial basada en el uso de dos PLDs, uno para la producción de los nuevos estados a partir de las entradas y el estado actual (PLD-1) y otro para producir las salidas a partir de las entradas y el estado (PLD-2). Este segundo circuito es de naturaleza combinacional y no usa los biestables D de sus macroceldas de salida, por lo que no vamos a detenernos ahora en su diseño.

Vamos a centrarnos en el primer PLD que es el que se encarga de la síntesis de las transiciones entre estados. Es decir, de la componente secuencial del autómata. Su arquitectura es la que reproducimos en la figura 9.12. Por razones pedagógicas hemos separado los retardos de la parte combinacional previa pero ya sabemos que si usamos un PLD tipo PALCE-22V10 descrito en el tema 7, sus macroceldas de salida (figura 7.31) incluyen un biestable D y dos multiplexos para controlar la salida al exterior del circuito y la realimentación. Suponemos entonces las configuraciones de control adecuadas en esos multiplexos (00 en el primero y 0 en el segundo) para que lo que sale de la PAL es el estado Q del biestable 0 y, a la vez, se realimenta para presentar a la entrada  $Q$  y  $\bar{Q}$ , junto a otras entradas externas y otras realimentaciones del resto de las macroceldas.

En estas condiciones la PAL queda configurada como se muestra en la parte (b) de la figura 9.12 y podemos empezar la descripción de los procesos de representación, análisis y síntesis de autómatas modulares con M variables de entrada ( $x_0, x_1, \dots, x_{M-1}$ ) y N variables de estado, ( $Q_0, Q_1, \dots, Q_{N-1}$ ) correspondientes a las salidas de los biestables D de N macroceldas de salida de la PAL.

Como tenemos M variables externas y, en general también tenemos la realimentación de las N macroceldas, las funciones de excitación de los N biestables D son funciones universales de (N+M) variables lógicas:

$$D_k(t_n) = f_k[x_0(t_{n-1}), \dots, x_{M-1}(t_{n-1}), Q_0(t_{n-1}), \dots, Q_{N-1}(t_{n-1})] \quad [9.14]$$
$$k = 0, 1, \dots, N-1$$



**Figura 9.12.** Síntesis con PLD's y biestables D. Arquitectura general mostrando la yuxtaposición de N autómatas universales de 2 estados para sintetizar un autómata de 2N estados y M variables de entrada. Las únicas entradas externas son las variables  $\{x_0, x_1, \dots, x_{M-1}\}$  y los coeficiente de control de los términos mínimos. **(a)** Aspecto global. **(b)** Detalle.

Cada biestable D (cada macrocelda) representa una variable de estado, de forma que la PAL es de hecho un procesador paralelo en el que la síntesis de un autómata de  $2^N$  estados se realiza yuxtaponiendo  $N$  autómatas de dos estados, uno por cada bit de la palabra que representa el estado. La programación del autómata se realiza cargando en el PLD la configuración de coeficientes  $\{a_i\}$  correspondientes a los términos mínimos que participan (para un autómata concreto) en las funciones de excitación,  $D_k(t_n)$ , de los biestables D de sus  $N$  macroceldas de salida. La reprogramación del autómata (su reconfiguración) puede realizarse cuantas veces sea necesario cargando los nuevos valores de los coeficientes  $\{a_i\}$  desde una EEPROM o desde un procesador externo a la memoria interna del PLD, tal como veremos en el tema 13 al estudiar los CPLDs (PLD complejos) y los FPGAs reconfigurables mediante memorias RAM.

### 9.3.1. Representación

Corresponde a los pasos generales P.1 y P.2. Dado un problema, obtenemos la descripción del circuito que necesitaríamos para su solución en términos del número de configuraciones de entrada necesarias,  $\{X_m\}$ , ( $m = 0, 1, \dots, 2^M-1$ ), del número de estados necesarios,  $\{S_i\}$  ( $i = 0, 1, \dots, 2^N-1$ ), y de las transiciones entre estos estados para cada uno de los valores mutuamente exclusivos de las configuraciones de entrada  $X_m$ . Es decir, buscamos reglas de la forma: ( $S_i$ , pasa a  $S_j$  bajo  $X_m$ ),

$$S_i \xrightarrow{X_m} S_j \quad [9.15]$$

Supuesto que se ha realizado la minimización (paso P.3) y dado que ya hemos elegido biestables D (paso P.4) y que la asignación de estados es automática (paso P.5), a través de la codificación en binario del subíndice, sólo nos falta obtener las matrices de transición de estados,  $\{T_{ij}^m\}$ , (paso P.2), que constituyen la representación formal del problema en este procedimiento de síntesis. Como para  $A$  estados necesitamos  $N$  biestables ( $2^N \geq A$ ), las matrices de transición serán de dimensión  $2^N \cdot 2^N$ . Como hemos necesitado  $\{X_m\}$  configuraciones de entrada ( $m = 0, 1, \dots, 2^M-1$ ), necesitaremos para su síntesis  $M$  variables lógicas ( $x_0, x_1, \dots, x_{M-1}$ ) y por consiguiente tendremos, en general,  $2^M$  matrices de transición distintas, una para cada configuración de entrada.

Los elementos de estas matrices booleanas,  $\{T_{ij}^m\}$ , son  $t_{ij}^m = 1$  si bajo  $X_m$  el autómata pasa de  $S_i$  a  $S_j$  y  $t_{ij}^m = 0$  si no se produce esa transición. Como bajo cualquier entrada  $X_m$ , el autómata tiene que

realizar alguna transición, estas matrices sólo tienen un uno por fila<sup>(\*)</sup>. Además como las configuraciones de entrada son distintas y mutuamente exclusivas, podemos usar una representación más compacta mediante una matriz paramétrica, llamada *matriz funcional*,  $M(X_m)$ , resultado de multiplicar cada matriz de transición,  $\{T_{ij}^m\}$ , por la configuración de entrada que la produce,  $(X_m)$ , y sumar estos productos. Así, la matriz funcional será:

$$M(X_m) = \sum_{m=0}^{2^M-1} T_{ij}^m \cdot X_m \quad [9.16]$$

Obsérvese que esta matriz es una matriz de representación. Es decir, es una forma compacta de representar el conjunto de expresiones lógicas que controlan todas y cada una de las transiciones de estado para todas y cada una de las posibles configuraciones de entrada. Cada configuración de valores en las variables de entrada,  $X_m$ , selecciona una matriz de transición,  $\{T_{ij}^m\}$ , que describe los cambios de estado que se producen cuando dejamos esa configuración de entrada constante. La *figura 9.13*. resume el proceso de representación.

Para centrar ideas pensemos cómo serían las matrices de transición para el caso mínimo de un autómata con sólo dos estados ( $S_0, S_1$ ). Si sólo tiene dos estados las posibles transiciones podrán describirse usando matrices booleanas 2x2 y sólo pueden haber cuatro matrices de este tipo que sean distintas, en función de dónde tienen el uno en cada una de sus dos filas:

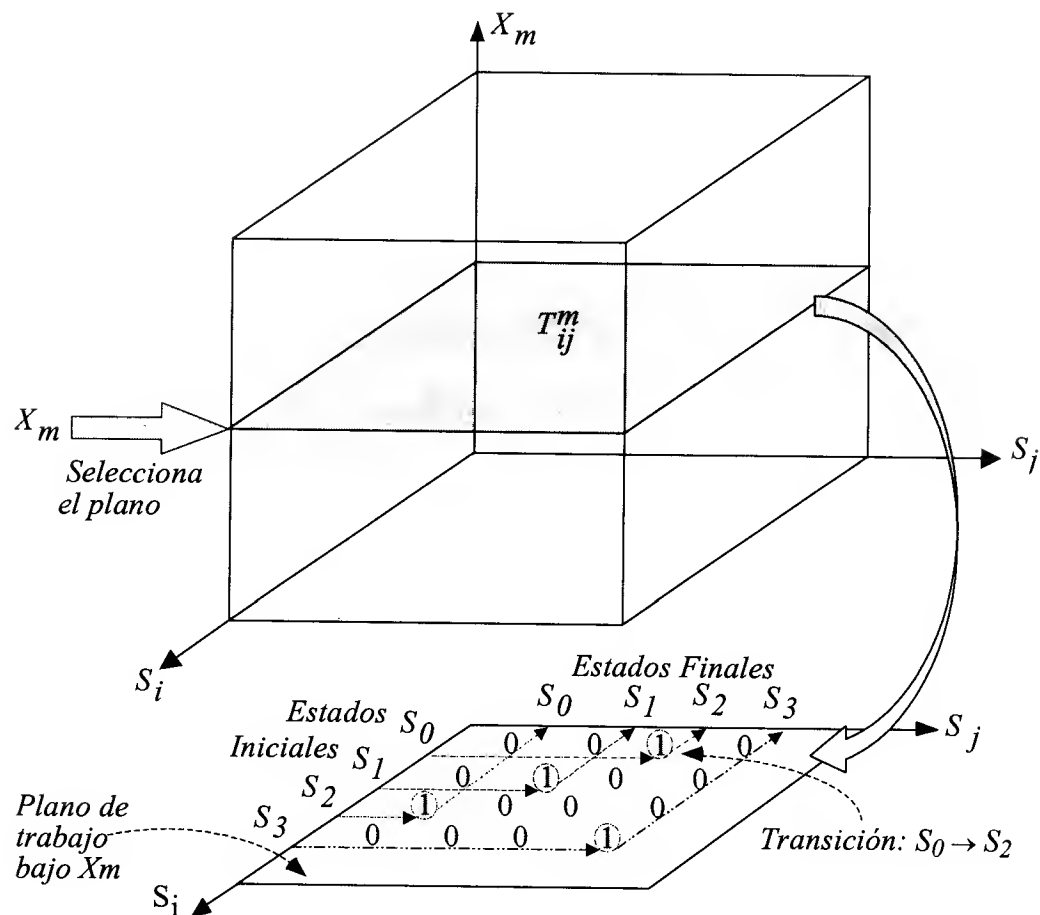
$$\begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \quad [9.17]$$

Como hay cuatro matrices, para su selección necesitamos dos variables de entrada ( $x_1, x_0$ ), de forma que ante cada configuración de entrada (00, 01, 10, 11) se activa una de las matrices de transición del autómata ( $T^{00}, T^{01}, T^{10}, T^{11}$ ). Una posible asociación sería:

---

(\*) Estamos hablando aquí sólo del caso determinísticos. Para circuitos probabilísticos o de lógica borrosa ("fuzzy"), los elementos de la matriz podrían tomar valores entre cero y uno.

$$T^{00} = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}, \quad T^{01} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad T^{10} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad T^{11} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \quad [9.18]$$



**Figura 9.13.** Representación por matrices de transición booleanas. Selección de plano y transiciones bajo entrada constante.

Evidentemente, cualquier otra forma de asociar configuraciones de entrada de las matrices de transición es también posible. Incluso, en muchas ocasiones tendremos más configuraciones que matrices o al revés, con lo que tendremos que asociar la misma matriz de transición a varias configuraciones de entrada. Si por el contrario tenemos menos configuraciones que matrices, cada forma de asociación dará lugar a un autómata diferente.

Supongamos en este ejemplo de cuatro matrices que sólo necesitamos una variable externa,  $x_0$ . Tenemos entonces dos configuraciones de entrada posibles ( $x_0 = 0$ ,  $x_0 = 1$ ) de modo que hay tantos autómatas distintos de dos estados y una entrada como formas distintas de asociar dos configuraciones con cuatro matrices. Por ejemplo, cogiendo las dos primeras, tendríamos:

$$\bar{x}_0 \Rightarrow \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}, \quad x_0 \Rightarrow \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad [9.19]$$

Y su matriz funcional será:

$$M(x_0) = \bar{x}_0 \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} + x_0 \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & \bar{x}_0 + x_0 \\ x_0 & \bar{x}_0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ x_0 & \bar{x}_0 \end{pmatrix} \quad [9.20]$$

Si quisiéramos un autómata que usara todas las matrices de transición posibles para dos estados, como hay cuatro matrices de transición diferentes, necesitaremos cuatro configuraciones de entrada, ( $X_m$ ), para seleccionarlas. Las generamos con dos variables ( $x_0$ ,  $x_1$ ) y obtenemos así el *autómata universal* de dos estados cuya matriz funcional es

$$M_{ij}^m(x_0, x_1) = \bar{x}_0 \cdot \bar{x}_1 \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} + \bar{x}_0 \cdot x_1 \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} + x_0 \cdot \bar{x}_1 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + x_0 \cdot x_1 \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \quad [9.21]$$

$$M_{ij}^m(x_0, x_1) = \begin{pmatrix} x_0 \cdot \bar{x}_1 + x_0 \cdot x_1 & \bar{x}_0 \cdot \bar{x}_1 + \bar{x}_0 \cdot x_1 \\ \bar{x}_0 \cdot x_1 + x_0 \cdot x_1 & \bar{x}_0 \cdot \bar{x}_1 + x_0 \cdot \bar{x}_1 \end{pmatrix} = \begin{pmatrix} x_0 & \bar{x}_0 \\ x_1 & \bar{x}_1 \end{pmatrix} \quad [9.22]$$

Para su síntesis necesitaríamos una PLD con dos entradas externas, ( $x_1$ ,  $x_0$ ) y una macrocelda de salida que realmente el estado (Q) de su biestable D hacia la parte combinacional de entrada al PLD.

### 9.3.2. Síntesis

Si tenemos la *representación* de un autómata de  $2^N$  estados y  $2^M$  configuraciones de entrada,  $\{X_m\}$ , el siguiente paso es la obtención de las funciones de excitación,  $\{D_k(t)\}$ , de los  $N$  biestables D que sintetizan al autómata. El algoritmo de síntesis y análisis de autómatas modulares se debe a R. Moreno Díaz y es el siguiente,

$$D_k(t) = \sum_{i=0}^{2^N-1} \left\{ \sum_j M_{ij}(X_m) \cdot S_j \right\} = \sum_{i=0}^{2^N-1} \sum_j M_{ij}(X_m) \cdot Q_0^a \cdot Q_2^b \cdot \dots \cdot Q_{N-1}^f. \quad [9.23]$$

La suma sobre el subíndice  $j$  se extiende a todos los estados,  $S_j$ , pertenecientes al subconjunto de estados,  $S_k$ , en los que la salida del biestable correspondiente,  $Q_k$ , está en alta,

$$j \mid S_j \in \{S_k \mid Q_k = 1\} \quad [9.24]$$

El estado inicial,  $S_i = Q_0^a \cdot Q_1^b \cdot \dots \cdot Q_{N-1}^f$ , recorre todos los valores posibles de  $i$ , desde cero hasta  $2^N-1$  puesto que en principio desde cualquier estado inicial se puede llegar a un estado final,  $S_j$ . Los exponentes de las variables de estado ( $a, b, \dots, f$ ) son las codificaciones en binario del estado  $S_i$ . Conviene recordar que hemos usado la notación de Gilstrap en la que:

$$Q_i^a = Q_i \quad \text{si} \quad a = 1 \quad \text{y} \quad Q_i^a = \bar{Q}_i \quad \text{si} \quad a = 0 \quad [9.25]$$

Vamos a ilustrar el proceso de síntesis con algunos ejemplos sencillos para autómatas de dos y cuatro estados. Más adelante usaremos de nuevo el método para el diseño de *contadores*.

Finalmente, en el último tema, dedicado a la lógica secuencial programable, haremos de nuevo referencia a estos procedimientos de síntesis. El estado actual de la tecnología permite diseñar autómatas finitos leyendo en una tabla LUT (Look Up Table) el estado sucesor del estado actual.

Empecemos sin embargo estudiando la síntesis de ejemplos muy sencillos para captar la forma de proceder del método. Supongamos que queremos sintetizar un autómata de dos estados ( $Q_0 = 0$ ,  $Q_0 = 1$ ) y una entrada ( $x_0$ ), cuya matriz funcional es:

$$M(x_0) = \begin{array}{c} \begin{array}{cc} Q_0=0 & Q_0=1 \end{array} \\ \begin{array}{c} Q_0=0 \\ Q_0=1 \end{array} \left( \begin{array}{cc} 0 & 1 \\ x_0 & \bar{x}_0 \end{array} \right) \end{array}$$

[9.26]

Es decir, si está en  $Q_0=0$ , al siguiente pulso de reloj pasa a  $Q_0=1$ , independientemente del valor de la entrada. En cambio, si está en  $Q_0=1$ , pasará a  $Q_0=0$  si la entrada está en alta ( $x_0=1$ ) ó se quedará

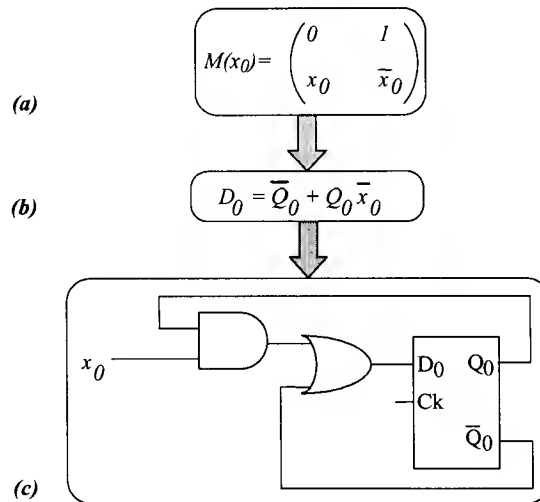


en  $Q_0=1$  si la entrada está en baja. Obsérvese cómo la suma de todos los elementos de una fila de la matriz debe ser igual a uno ( $0+1=1$ ,  $x_0+\bar{x}_0=1$ ). Las configuraciones de entrada son mutuamente exclusivas y a algún estado final habrá que llegar. Veamos ahora los caminos por los que se puede llegar a cada uno de los estados finales posibles.

Los caminos que llegan a  $Q_0 = 1$  son los de la segunda columna. Es decir, cuando el estado anterior era  $Q_0 = 0$  para cualquier valor de la entrada y cuando el estado anterior es  $Q_0 = 1$ , para  $x_0 = 0$  ( $\bar{x}_0$ ). Por consiguiente, la función de excitación será:

$$D_0 = \bar{Q}_0 \cdot 1 + Q_0 \cdot \bar{x}_0 = \bar{Q}_0 + Q_0 \cdot \bar{x}_0 \quad [9.27]$$

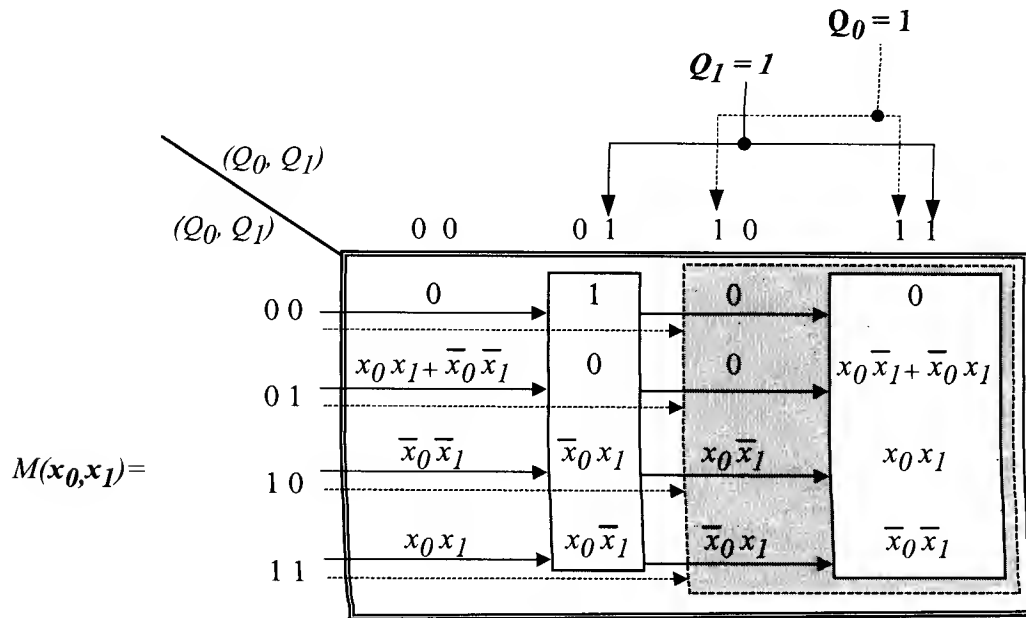
El circuito correspondiente se ilustra en la *figura 9.14*.



**Figura 9.14.** Ilustración del proceso de síntesis. (a) Representación. (b) Cálculo de la función de excitación. (c) Circuito.

Supongamos ahora que tras los pasos P.1 y P.2 del procedimiento general hemos obtenido la matriz funcional que representa a un autómata de cuatro estados (codificados con dos biestables,  $Q_0$  y  $Q_1$ ) y cuatro configuraciones de entrada, ( $X_0, X_1, X_2, X_3$ ), codificadas con dos variables ( $x_0, x_1$ ) cada uno de cuyos términos mínimos representa una configuración ( $x_0=0, x_1=0$ :  $X_0$ ;  $x_0=0, x_1=1$ :  $X_1$ ;  $x_0=1, x_1=0$ :  $X_2$ ;  $x_0=1, x_1=1$ :  $X_3$ ). La *figura 9.15* muestra la matriz funcional de este autómata y la 9.16 el circuito correspondiente usando una PROM de 4 entradas.

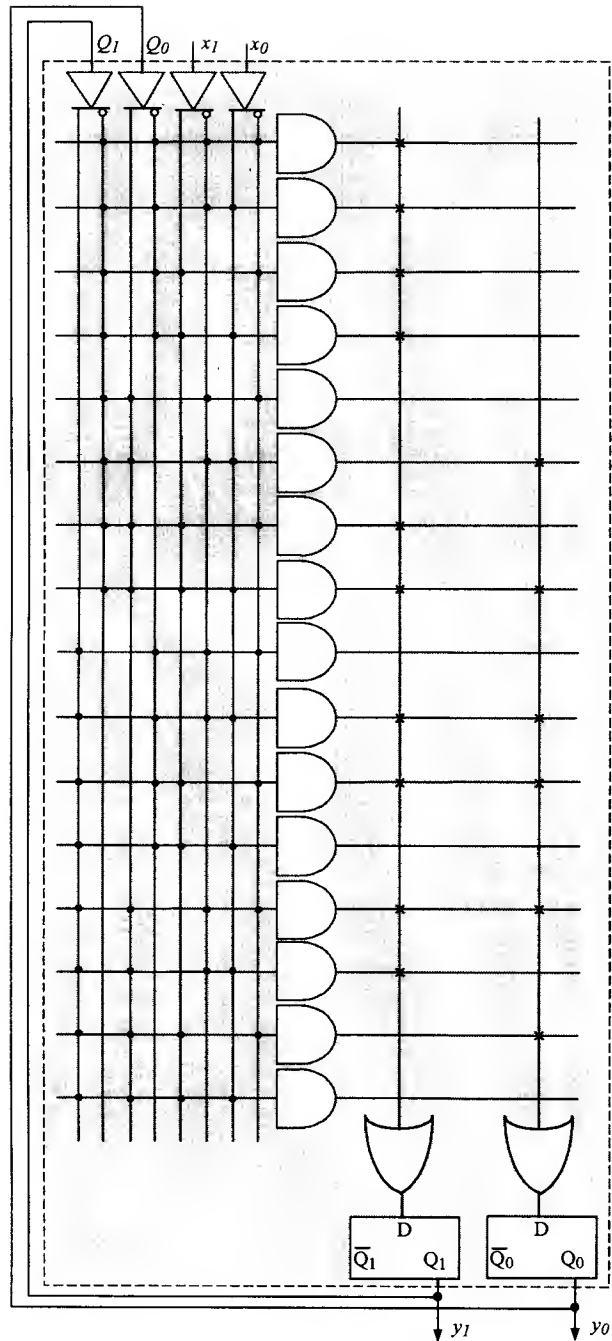
Para facilitar la comprensión del proceso de síntesis dibujamos los estados iniciales y finales y los caminos que llegan a cada estado final. De acuerdo con el algoritmo, se clasifican los estados finales en tres grupos (aquellos que tienen  $Q_0=1$ , los que tienen  $Q_1=1$  y el estado 00, que tiene ambos biestables a cero).



**Figura 9.15.** Matriz funcional correspondiente a un autómata de cuatro estados y dos variables lógicas de entrada. Los estados se codifican con dos biestables D y en cada una de las 16 posiciones de la matriz aparece el término mínimo o la suma de términos mínimos que provoca la transición entre la fila y la columna correspondientes.

Obsérvese que la suma de cualquier fila de la matriz es 1. Cuando aparece  $x_0$  y  $x_1$ , deben aparecer todos los términos mínimos en una u otra posición de la fila. Cuando aparece un uno, el resto de los elementos de la fila deben ser necesariamente cero.

Veamos ahora el cálculo de las funciones de excitación  $D_0$  y  $D_1$ . Para calcular la expresión de  $D_0$  hay que sumar todos los caminos que llevan desde cualquier estado inicial a los estados finales (10 ó 11). Es decir a alguno de los estados finales en los que  $Q_0=1$ . En el ejemplo de la figura 9.15 sumaremos las columnas tercera y cuarta multiplicando la configuración de cada uno de los estados iniciales por los elementos correspondientes de la matriz.



**Figura 9.16.** Síntesis de autómatas finitos de cuatro estados y dos variables externas con una arquitectura PROM de cuatro entradas.

Así pues, la expresión de  $D_0$  será:

$$D_0 = \overline{Q_0} \cdot Q_1 \left( \overbrace{x_0 \cdot \bar{x}_1 + \bar{x}_0 \cdot x_1}^{x_0} \right) + Q_0 \cdot \overline{Q_1} \left( \overbrace{x_0 \cdot \bar{x}_1 + x_0 \cdot x_1}^{\bar{x}_0} \right) + Q_0 \cdot Q_1 \left( \overbrace{\bar{x}_0 \cdot x_1 + \bar{x}_0 \cdot \bar{x}_1}^{\bar{x}_0} \right) \quad [9.28]$$

Como se puede ver el estado inicial 00 ( $\overline{Q_0}, \overline{Q_1}$ ) no interviene porque los dos elementos correspondientes de la matriz que conducen a los estados finales 10 y 11 son 0. Análogamente el estado inicial 01 ( $\overline{Q_0}, Q_1$ ) interviene multiplicado por el elemento de la matriz ( $x_0 \cdot \bar{x}_1 + \bar{x}_0 \cdot x_1$ ) que es el que controla la transición hacia el estado final 11 en el que  $Q_0 = 1$  y así sucesivamente para los otros dos estados iniciales 10, 11.

Procediendo de la misma forma, para obtener  $D_1$  sumamos los caminos que llevan desde cualquier estado inicial a los dos estados finales que poseen el bit  $Q_1$  en alta (estados 01 y 11),

$$D_1 = \overline{Q_0} \cdot \overline{Q_1} + \overline{Q_0} \cdot Q_1 \left( \overbrace{x_0 \cdot \bar{x}_1 + \bar{x}_0 \cdot x_1}^{x_1} \right) + Q_0 \cdot \overline{Q_1} \left( \overbrace{\bar{x}_0 \cdot x_1 + x_0 \cdot x_1}^{\bar{x}_1} \right) + Q_0 \cdot Q_1 \left( \overbrace{x_0 \cdot \bar{x}_1 + \bar{x}_0 \cdot \bar{x}_1}^{\bar{x}_1} \right) \quad [9.29]$$

El circuito correspondiente se muestra en la *figura 9.16*.

Obsérvese que podríamos simplificar las expresiones ( $x_0 \cdot \bar{x}_1 + x_0 \cdot x_1 = x_0$ , etc...) pero no lo hacemos porque en síntesis con PROM disponemos de todos los términos mínimos y es más sencillo ver así, sin simplificar, cuáles de ellos participan en la función. Como consecuencia del procedimiento de obtención de  $D_0$  y  $D_1$ , generamos primero la parte de los términos mínimos correspondiente a las variables de estado ( $\overline{Q_0} \overline{Q_1}, \overline{Q_0} Q_1, Q_0 \overline{Q_1}, Q_0 Q_1$ ). Por otra parte, generamos las componentes de los términos mínimos de las variables de entrada ( $\bar{x}_0 \bar{x}_1, \bar{x}_0 x_1, x_0 \bar{x}_1, x_0 x_1$ ), y finalmente multiplicamos los que tengamos que multiplicar y sumamos sobre dos líneas OR los productos correspondientes a  $D_0$  y  $D_1$ . Los productos son siempre de la forma:

$(\text{Codificación binaria de } \mathbf{Sinicial}) \cdot (\text{Configuración } \mathbf{X_m} \text{ que lleva a } \mathbf{Sfinal})$

en el que la salida del biestable  $Q_i$  está en alta ( $i = 0, 1, \dots, N-1$ ).

### 9.3.3. Análisis

Ya hemos visto la representación y la síntesis. Veamos ahora el *análisis*. Es el proceso inverso al de síntesis. Ahora partimos de un circuito como el de la *figura 9.16* en el que aparecen, en general,  $N$  biestables  $D$  ( $Q_0, Q_1, \dots, Q_{N-1}$ ), y  $N$  funciones de excitación, ( $D_0, D_1, \dots, D_{N-1}$ ), y queremos obtener la *representación* del autómata de  $2^N$  estados correspondiente. Es decir, las matrices de transición  $\{T_{ij}^m\}$ , ó la matriz funcional  $M(X_m)$ .

Para que el autómata pase del estado  $S_i$  a  $S_j$ , cada una de sus variables de estado ( $Q_k$ ) deben pasar de cómo estaban en  $S_i$  a como están en  $S_j$ . Como esto debe ocurrir para todas las  $Q_k$  ( $k=0, 1, \dots, N-1$ ), la condición de transición aparece como un *producto de las condiciones de transición* individuales. Si ( $q, r, \dots, t$ ) son los dígitos binarios correspondientes al estado final  $S_j$ , entonces:

$$m_{ij}(X_m) = f_0^q(X_m; S_i) \cdot f_1^r(X_m; S_i) \cdot \dots \cdot f_{N-1}^t(X_m; S_i) \quad [9.30]$$

siguiendo de nuevo la notación de Gilstrap en la que

$$\begin{aligned} f_n^0(X; S_i) &= \overline{f_n(X; S_i)} \\ f_n^1(X; S_i) &= f_n(X; S_i) \end{aligned} \quad [9.31]$$

Para ilustrar el procedimiento general de análisis vamos a aplicarlo al mismo ejemplo que hemos usado en síntesis. Tenemos entonces como punto de partida el circuito de la *figura 9.8* y la expresión de las funciones de excitación, que repetimos aquí por comodidad,

$$D_0 = \bar{Q}_0 \cdot Q_1 (x_0 \cdot \bar{x}_1 + \bar{x}_0 \cdot x_1) + Q_0 \cdot (\bar{Q}_1 \cdot x_0 + Q_1 \cdot \bar{x}_0) \quad [9.32]$$

$$D_1 = \bar{Q}_0 \cdot \bar{Q}_1 + \bar{Q}_0 \cdot Q_1 (x_0 \cdot \bar{x}_1 + \bar{x}_0 \cdot x_1) + Q_0 (\bar{Q}_1 \cdot x_1 + Q_1 \cdot \bar{x}_1) \quad [9.33]$$

Calcularemos algunos términos de la matriz funcional para que, una vez familiarizado con el procedimiento, el alumno pueda calcular el resto, comprobando que coincide con la matriz de partida en el apartado de síntesis. La ecuación 9.30 puede volverse a escribir de forma más descriptiva como:

$$m_{ij} = D_0^{Q_0|S_{final}}(x_1, x_2; Q_0|_{Sinicial}, Q_I|_{Sinicial}) \cdot D_I^{Q_I|S_{final}}(x_1, x_2; Q_0|_{Sinicial}, Q_I|_{Sinicial}) \quad [9.34]$$

Así por ejemplo el elemento  $m_{21}$  que especifica la transición entre  $S_{inicial}$  ( $Q_0 = 1, Q_I = 0$ ) y  $S_{final}$  ( $Q_0 = 0, Q_I = 1$ ) será:

$$\begin{aligned} m_{21} &= D_0^0(x_1, x_2; Q_0|_{Sinicial}, Q_I|_{Sinicial}) \cdot D_I^1(x_1, x_2; Q_0|_{Sinicial}, Q_I|_{Sinicial}) = \\ &= \overline{D_0(x_1, x_2; Q_0|_{Sinicial}, Q_I|_{Sinicial})} \cdot D_I(x_1, x_2; Q_0|_{Sinicial}, Q_I|_{Sinicial}) \end{aligned} \quad [9.35]$$

Calculemos ahora los valores de  $D_0$  y  $D_I$  a partir de las ecuaciones del enunciado [9.31] y [9.32] en las que sustituimos los valores que toman  $Q_0$  y  $Q_I$  para el estado inicial. Así,

$$D_0 = \bar{1} \cdot 0(x_0 \cdot \bar{x}_1 + \bar{x}_0 \cdot x_1) + 1(\bar{0} \cdot x_0 + 0 \cdot \bar{x}_0) = 0 + x_0 = x_0 \quad [9.36]$$

$$D_I = \bar{1} \cdot \bar{0} + \bar{1} \cdot 0(x_0 \cdot \bar{x}_1 + \bar{x}_0 \cdot x_1) + 1(\bar{0} \cdot x_1 + 0 \cdot \bar{x}_1) = 0 + 0 + x_1 = x_1 \quad [9.37]$$

Por último deberemos sustituir estos valores en la expresión de  $m_{21}$  [9.35] y obtenemos:

$$m_{21} = \bar{D}_0 \cdot D_I = \bar{x}_0 \cdot x_1 \quad [9.38]$$

De la misma forma podemos calcular el elemento  $m_{30}$  que especifica la transición entre 11 y 00

$$\begin{aligned} m_{30} &= D_0^0(x_1, x_2; Q_0|_{Sinicial}, Q_I|_{Sinicial}) \cdot D_I^0(x_1, x_2; Q_0|_{Sinicial}, Q_I|_{Sinicial}) = \\ &= \overline{D_0(x_1, x_2; Q_0|_{Sinicial}, Q_I|_{Sinicial})} \cdot \overline{D_I(x_1, x_2; Q_0|_{Sinicial}, Q_I|_{Sinicial})} \end{aligned} \quad [9.39]$$

Calculemos ahora los valores de  $D_0$  y  $D_I$ .

$$D_0 = \bar{1} \cdot 1(x_0 \cdot \bar{x}_1 + \bar{x}_0 \cdot x_1) + 1(\bar{1} \cdot x_0 + 1 \cdot \bar{x}_0) = 0 + \bar{x}_0 = \bar{x}_0 \quad [9.40]$$

$$D_I = \bar{1} \cdot \bar{1} + \bar{1} \cdot 1(x_0 \cdot \bar{x}_1 + \bar{x}_0 \cdot x_1) + 1(\bar{1} \cdot x_1 + 1 \cdot \bar{x}_1) = 0 + 0 + \bar{x}_1 = \bar{x}_1 \quad [9.41]$$

$$\text{Por tanto} \quad m_{30} = \overline{D_0} \cdot \overline{D_I} = x_0 \cdot x_1 \quad [9.42]$$

Por último el elemento  $m_{00}$  es el término que especifica la transición entre

$$S_{inicial} (Q_0 = 0, Q_I = 0) \quad \text{y} \quad S_{final} (Q_0 = 0, Q_I = 0)$$

luego:

$$m_{00}(x_0, x_1) = \bar{D}_0 \cdot \bar{D}_1 \quad [9.43]$$

$$D_0 \quad (\text{para } Q_0 = 0 \quad \text{y} \quad Q_1 = 0) = 0, \Rightarrow D_0^0 = 0^0 = 1 \quad [9.44]$$

$$D_1 \quad (\text{para } Q_0 = 0 \quad \text{y} \quad Q_1 = 0) = 1, \Rightarrow D_1^0 = 1^0 = 0$$

$$\text{luego,} \quad m_{00} = 1 \cdot 0 = 0 \quad [9.45]$$

Como podemos ver los elementos calculados coinciden con los correspondientes elementos de la matriz de la *figura 9.15*. De igual forma se calcula el resto de los elementos.

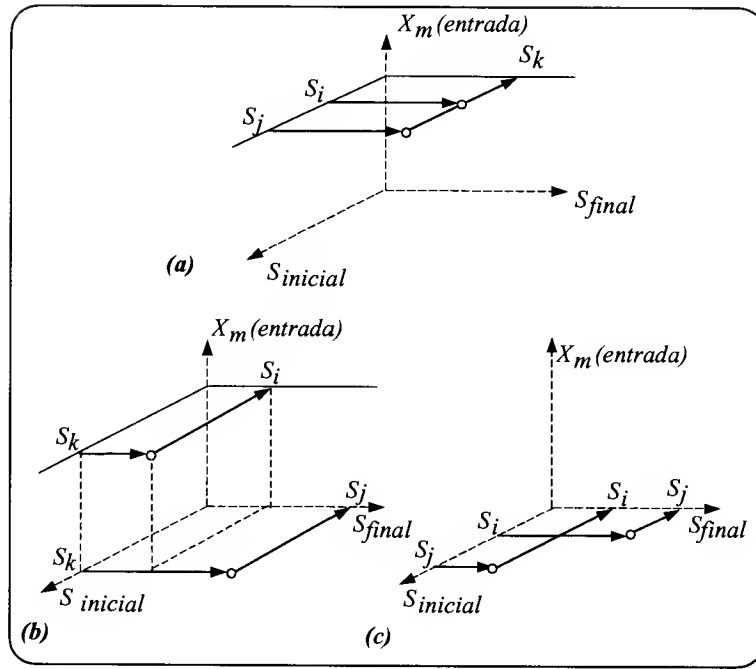
El siguiente paso sería, partiendo de la matriz funcional que hemos calculado, dibujar el diagrama de transición de estados ya que cada elemento de la matriz nos dice bajo qué configuración de entrada se produce la transición correspondiente.

## 9.4. DISEÑO CON BIESTABLES J-K

El diseño con biestables J-K sigue los mismos pasos generales hasta llegar al cálculo de las funciones de excitación de los biestables, sólo que ahora no es tan sencilla como en el caso D. Hay que recorrer los estados, desde  $S_0(0, 0, \dots, 0)$  hasta  $S_{2N-1}(1, 1, \dots, 1)$  calculando  $J_i$  y  $K_i$  para cada uno de los bits que codifican el estado, de acuerdo con las tablas de las *figuras 9.7.b* y *c*. Nosotros ilustraremos este procedimiento en los siguientes apartados dedicados al estudio de contadores y registros.

Aquí, a nivel general, sólo queremos comentar el problema de la reordenación de estados. Cuando la síntesis se hace con electrónica no programable, el problema de la asignación de estados es más complicado que la codificación binaria del subíndice usada a lo largo de todo este apartado sobre representación, análisis y síntesis ya que ahora conviene optimizar el diseño. Para ello, se reordenan los estados intentando conseguir que los unos y los ceros de las funciones de excitación de las entradas  $J$  y  $K$  de los  $N$  biestables queden juntos en los mapas de Karnaugh para obtener ecuaciones de disparo lo más simplificadas posible.

Para ello la reordenación se hace buscando "adyacencias" entre estados. Dos estados pueden ser adyacentes en uno o más de los siguientes sentidos (*figura 9.17*):



**Figura 9.17.** Distintas relaciones de adyacencia. **(a)** Relación de adyacencia  $A_1$  (estados iniciales distintos dan lugar al mismo estado final). **(b)** Un mismo estado inicial con distintas entradas da lugar a distintos estados finales  $A_2$ . **(c)**  $A_3$ , ciclo de oscilación.

1. Si para alguna entrada ambos estados tienen el mismo sucesor (relación de adyacencia  $A_1$ ). El estado  $S_i$  es adyacente a estado  $S_j$ , es decir, se cumple la relación de equivalencia,  $A_1(S_i, S_j)$ , si existen configuraciones de entrada  $X_m$  y estados  $S_k$  tales que

$$S_i \xrightarrow{X_m} S_k \quad \text{y} \quad S_j \xrightarrow{X_m} S_k \quad [9.46]$$

2. Si son sucesores de un mismo estado para entradas adyacentes (relación de adyacencia  $A_2$ ).

$$A_2(S_i, S_j) \Leftrightarrow \text{existe un } S_k \text{ y dos entradas } (X_m, X_{m+1}) \text{ tales que :} \\ S_k \xrightarrow{X_m} S_i \quad \text{y} \quad S_k \xrightarrow{X_{m+1}} S_j \quad [9.47]$$

3. Si cierran un ciclo de oscilación bajo entrada constante (relación de adyacencia  $A_3$ ).

$$A_3(S_i, S_j) \Leftrightarrow \text{existe una entrada } X_m \text{ tal que : } S_i \xrightarrow{X_m} S_j \quad \text{y} \quad S_j \xrightarrow{X_m} S_i \quad [9.48]$$



4. Si existe alguna entrada para la que ambos estados den la misma salida (relación de adyacencia  $A_4$ )

$$A_4(S_i, S_j) \Leftrightarrow X_m \text{ tal que: } g(S_i, X_m) = g(S_j, X_m) \quad [9.49]$$

Lógicamente, cada regla de adjudicación de adyacencias entre estados ( $A_1$ ,  $A_2$ ,  $A_3$  y  $A_4$ ) introduce un conjunto de restricciones de forma que es prácticamente imposible que un circuito secuencial real pueda satisfacerlas todas. Un procedimiento simplificado y razonable es dar prioridad a aquellas adyacencias que cumplan las siguientes reglas:

**Regla A.1 generalizada.** Se deben dar asignaciones adyacentes a estados con igual estado siguiente para alguna entrada. Cuanto mayor sea el número de estados siguientes iguales, mayor prioridad.

**Regla A.2 generalizada.** Se deben dar asignaciones adyacentes a estados sucesores de un mismo estado para entradas adyacentes (que sólo varían en el valor de uno de sus bits).

Como en el resto de la electrónica digital, al considerar las posibles soluciones a un problema concreto hay que empezar por analizar los datos de catálogo para ver si existe la solución directa ya integrada. Sólo en el caso que no exista, tendrá sentido pasar a su síntesis con PLAs que ya incluyen biestables tipo D en sus macroceldas de salida, de acuerdo con el procedimiento general descrito anteriormente y que resumimos aquí en los cuadros de la *figura 9.18*.

Terminamos así la primera parte de este capítulo encaminada a la introducción del diseño secuencial en general. Vamos a presentar ahora la segunda parte dedicada al estudio de dos tipos concretos de circuitos secuenciales: los contadores y los registros. En ambos casos, utilizaremos la teoría general para introducir la función y después bajaremos a realizaciones específicas integradas en MSI y diseñadas a partir de biestables J-K. En estos casos junto a la función específica de "contar" ó almacenar y/o desplazar", existen otras facilidades de establecimiento de condiciones iniciales (*preset* y *clear*), de comportamiento síncrono o asíncrono y de selección de circuito y conexión en cascada.

## ANÁLISIS

**Matriz Funcional:**  $M(X_m)$

$$m_{ij}(X_m) = f_0^q(X_m; S_i) \cdot f_1^r(X_m; S_i) \cdots f_{N-1}^t(X_m; S_i)$$

siendo:  $m_{ij}(X_m)$  los elementos de la matriz funcional

$(q, r, \dots, t)$  los dígitos binarios correspondientes al estado final  $S_j$

$S_i$  = estado inicial       $S_j$  = estado final

**Notación de Gilstrap:**

$$f_n^0(X_m; S_i) = \overline{f_n(X_m; S_i)} \quad f_n^1(X_m; S_i) = f_n(X_m; S_i)$$

## SÍNTESIS

**Funciones de Excitación de los N biestables D:**  $D_k(t)$

$$D_k(t) = \sum_{i=0}^{2^{N-1}} \left\{ \sum_j M_{ij}(X_m) \cdot S_j \right\} = \sum_{i=0}^{2^{N-1}} \sum_j M_{ij}(X_m) \cdot Q_0^a \cdot Q_1^b \cdots Q_{N-1}^f$$

para todo  $j | S_j \{ S_k | Q_k = 1 \}$

$$S_i = Q_0^a \cdot Q_1^b \cdots Q_{N-1}^f = \text{estado inicial}$$

$(a, b, \dots, f)$  codificación binaria correspondientes al estado inicial  $S_i$

$S_j$  = estado final

**Notación de Gilstrap:**

$$Q_i^a = Q_i \quad \text{si} \quad a = 1$$

$$Q_i^a = \overline{Q_i} \quad \text{si} \quad a = 0$$

**Figura 9.18.** Resumen de los algoritmos de análisis y síntesis de autómatas finitos.

## 9.5. CONTADORES

Los contadores son circuitos secuenciales capaces de recorrer una secuencia previamente especificada de estados. En general reciben como entrada un tren de impulsos y responden con una sucesión de estados correspondientes a la representación en binario del número de impulsos recibidos

desde que se inició el ciclo. Por eso se les llama contadores.

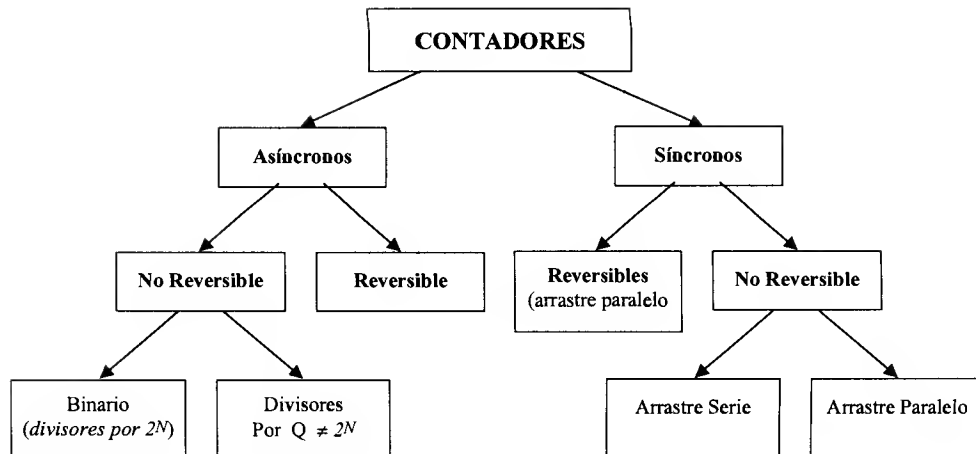
Formalmente los contadores son autómatas finitos de  $2^N$  estados en los que, en general, la asignación de estados se corresponde con la codificación en binario del cardinal correspondiente al estado. Así, un contador de 3 bits es un autómata de  $2^3=8$  estados ( $S_0, S_1, \dots, S_7$ ) y una entrada externa  $x(t)$ , tal que situado el contador en el estado inicial  $S_0(000)$ , va pasando sucesivamente por los estados  $S_1(001)$ ,  $S_2(010)$ , etc. cada vez que llega un pulso a la entrada, hasta alcanzar el estado  $S_7(111)$ , en el que finaliza el ciclo. Si, estando en este estado llega otro pulso, el contador vuelve al estado inicial,  $S_0(000)$ .

Si pensamos ahora en la estructura interna de los contadores ya sabemos que para sintetizar un autómata con un número  $A$  de estados necesitamos  $N$  biestables siendo  $2^N \geq A$ . Es decir, con  $N$  biestables podemos diseñar contadores desde 0 hasta  $2^N-1$ .

Los biestables pueden ser R-S, T, D ó J-K y nosotros haremos énfasis en el diseño con J-K y D. En el primer caso porque los R-S y T pueden obtenerse como casos particulares y porque los J-K permiten un mejor acomodo de un conjunto de señales de control complementarias para establecer un cierto estado inicial. Además, el diseño con J-K es el estándar en la integración en media escala (MSI). Haremos énfasis también en el diseño usando biestables D porque nos permiten un uso inmediato del procedimiento general de síntesis de cualquier autómata usando lógica programable.

Para un mismo modelo matemático soporte existen distintos tipos de contadores que se distinguen por la forma de transferir los datos entre biestables vecinos y la lógica adicional de establecimiento de estados específicos, carga o descarga paralelo, y el carácter unidireccional o reversible del ciclo de contar. El diagrama de la *figura 9.19*. presenta una primera clasificación.

Esta clasificación hace referencia a la forma general de organización del sistema digital en el que se va a integrar el contador (síncronos o asíncronos) y a la lógica de control complementaria que incorpora. Las primeras soluciones integradas eran asíncronas, porque necesitaban menos puertas de control para transferir los datos entre biestables pero también eran soluciones lentas. Para terminar la transición de estado en un contador de  $N$  bits, tenemos que esperar a que el biestable del bit  $(N-1)$  dé la orden de conmutar (o no conmutar) al biestable del bit  $N$ . Al igual que en las operaciones aritmético-lógicas, la solución es complicar la lógica combinacional de control de las entradas  $J$  y  $K$  ó  $D$  de cada uno de los biestables para que la conmutación de todos ellos (el paso del estado global  $S_i$  al  $S_j$ ) se haga a la vez. Así, el retardo global es el mismo que el del primer bit.



**Figura 9.19** Clasificación de los distintos tipo de contadores.

El carácter reversible o no reversible tiene que ver con la posibilidad de recorrer el ciclo de contar “*hacia arriba*” o “*hacia abajo*”, dependiendo del valor de una variable de control. Es decir, de pasar de  $S_1=001$  a  $S_2=010$  ó al revés. Piense por ejemplo en un contador asíncrono encargado de calcular cuantos coches hay en un aparcamiento. Cada vez que entra un coche nuevo, incrementa su estado y cada vez que sale uno lo decremента. Cuando no hay movimiento retiene el último estado.

Decimos que un contador asíncrono es binario cuando le dejamos terminar su ciclo máximo ( $2^N$  para  $N$  bits). En cambio, decimos que es un divisor por  $Q$  (menor y distinto de  $2^N$ ) cuando se corta el ciclo de incrementar el contenido del contador en ese valor  $Q$  (5, 7, 12, etc.), devolviendo desde aquí al contador a su estado inicial (00...0).

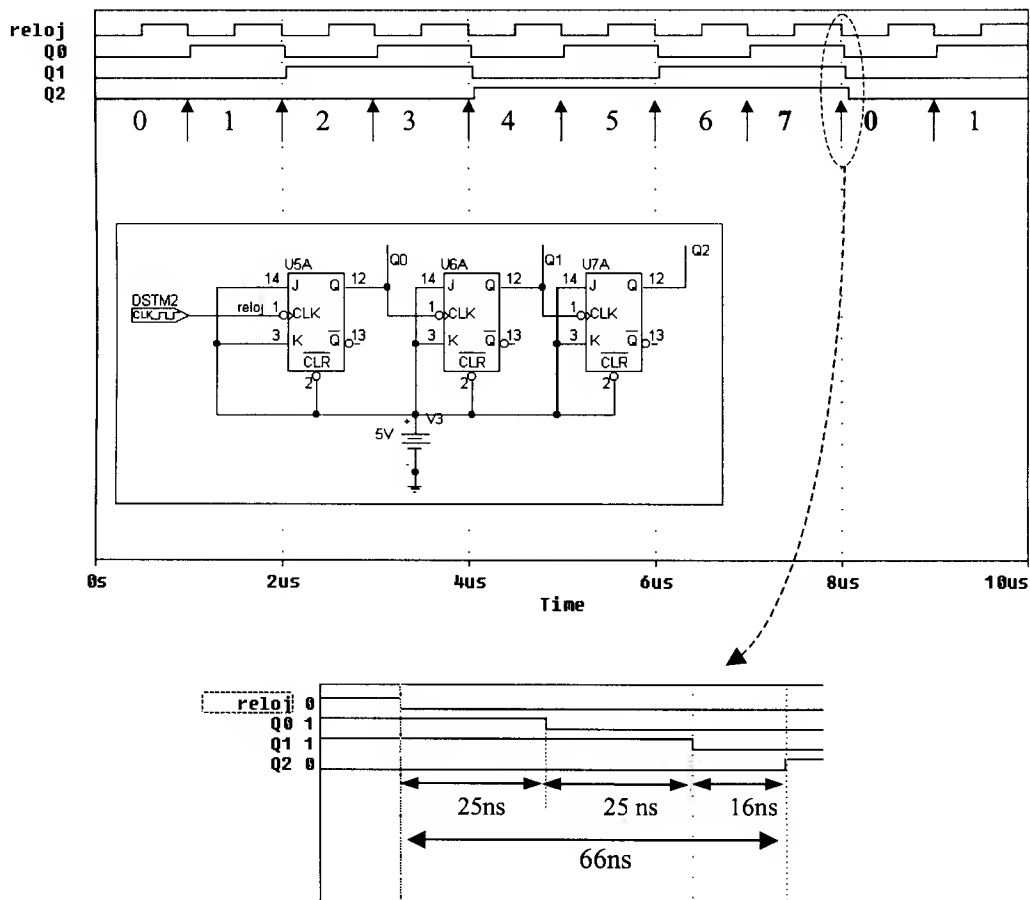
Finalmente, dentro de los síncronos, el arrastre puede ser serie o paralelo, dependiendo de la lógica combinatorial complementaria. Lógicamente, los de arrastre paralelo son más rápidos.

Veremos ahora la estructura interna de los distintos tipos de contadores en orden de dificultad creciente y después ilustraremos el procedimiento general de síntesis con PLDs con macroceldas de salida que incluyen biestables D.

### 9.5.1. Contadores Asíncronos

La forma más sencilla de diseñar un contador asíncrono binario de tres bits, por ejemplo, es usar tres biestables J-K con  $J=K=1$  (es decir, actuando como T y con  $T=1$ ) y usar la entrada de reloj

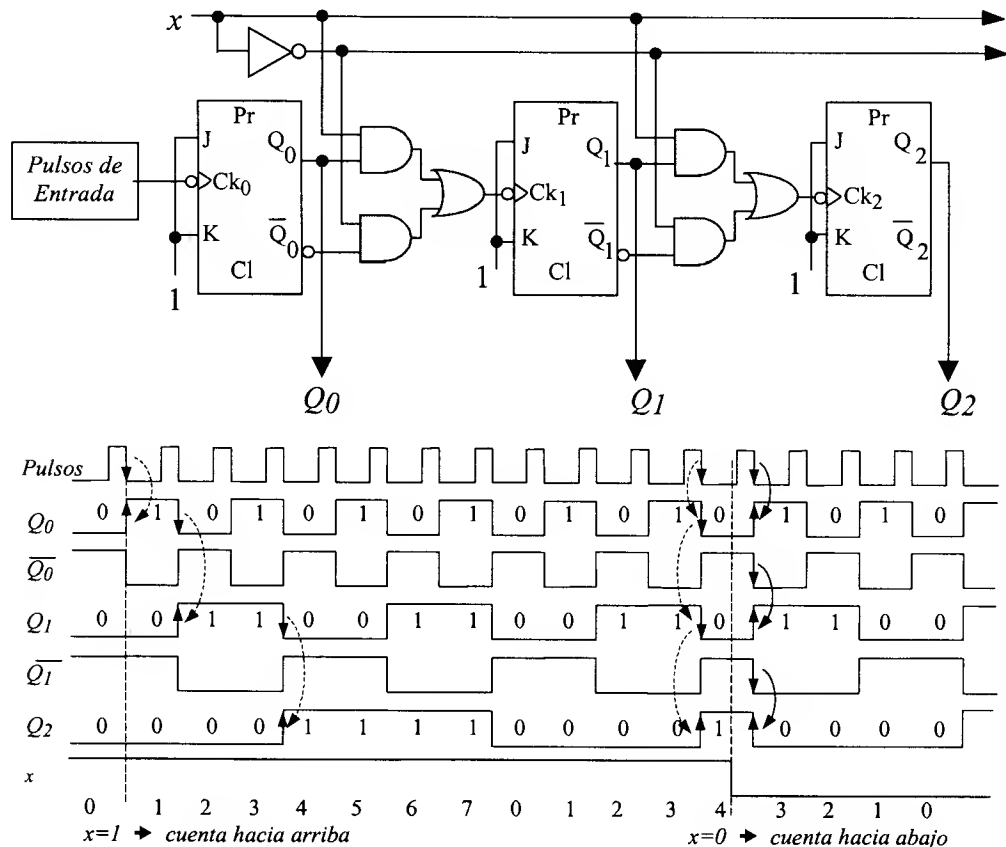
como entrada general al contador, es decir como variable lógica cuyo número de impulsos se desea contar. Después, la salida  $Q$  del primer biestable ( $Q_0$ ) se conecta a la entrada de reloj del segundo y así sucesivamente. Si suponemos que los biestables J-K usados son de los que se disparan con la bajada del pulso de reloj (transición de alta a baja), el circuito del contador y el cronograma correspondiente son los que se muestran en la *figura 9.20*.



**Figura 9.20.** Contador asíncrono de 3 bits construido con J-K de disparo a bajadas.

Obsérvese que cada estado se retiene hasta la llegada del siguiente pulso de reloj y que estos pulsos no tienen por qué estar llegando constantemente. Cuando llega un nuevo pulso, a su bajada se produce el cambio de estado.

Si repetimos ahora el cronograma equivalente al de la *figura 9.20* pero incluyendo también la evolución temporal de las salidas  $\bar{Q}_0$  y  $\bar{Q}_1$  de los biestables, vemos que si usamos estas señales  $\bar{Q}$  para disparar al biestable siguiente, en vez de usar las  $Q$ , ahora el circuito cuenta “*hacia abajo*”, recorriendo la secuencia 7-6-5-4-3-2-1-0. Por consiguiente, es fácil obtener un contador asíncrono reversible, si multiplexamos el acarreo a la entrada de reloj bajo el control de una nueva variable externa,  $x$ , de forma que para  $x=1$  (contar “*hacia arriba*”) facilita el paso de  $Q_0$  y  $Q_1$  al reloj del segundo y tercer biestable, respectivamente. Inversamente, para  $x=0$  (contar “*hacia abajo*”), el multiplexo facilita el acceso a las variables negadas ( $\bar{Q}_0$  hacia el reloj  $Ck_1$  y  $\bar{Q}_1$  hacia el reloj  $Ck_2$ ). La *figura 9.21* muestra el circuito correspondiente y los cronogramas para  $x=0$  y  $x=1$ .



**Figura 9.21.** Contador asíncrono reversible de 4 bits.

En el caso  $x=0$ , evidentemente, tenemos el mismo resultado que en la *figura 9.20*. En cambio,

al tomar las señales de  $\overline{Q}_0$  y  $\overline{Q}_1$ , la secuencia que sigue es la inversa. Pasa de 7 a 6 a 5 ... a 1 y a 0.

Obviamente, la línea de control “up/down” ( $x$ ) no debe de cambiar su valor entre pulsos de entrada, porque se podrían producir cuentas erróneas debido a esas transiciones. Veremos a continuación que los contadores síncronos no presentan esa dificultad y por eso los contadores reversibles suelen ser síncronos.

Véase cómo dependiendo del sentido en el que queremos contar tomamos las señales de  $Q$  o de  $\overline{Q}$ . El cronograma adjunto muestra las relaciones causa-efecto entre las distintas transiciones empezando siempre con el paso de alta a baja de los pulsos del reloj. En la primera parte del cronograma el contador cuenta hacia arriba, por lo que se consideran las transiciones de alta a baja en  $Q_0$ ,  $Q_1$  y  $Q_2$ . A partir de un cierto punto, cambia el valor de  $x$  y empieza a contar hacia arriba. A partir de ese momento, las transiciones que se consideran son las de  $\overline{Q}_0$ ,  $\overline{Q}_1$  y  $\overline{Q}_2$ .

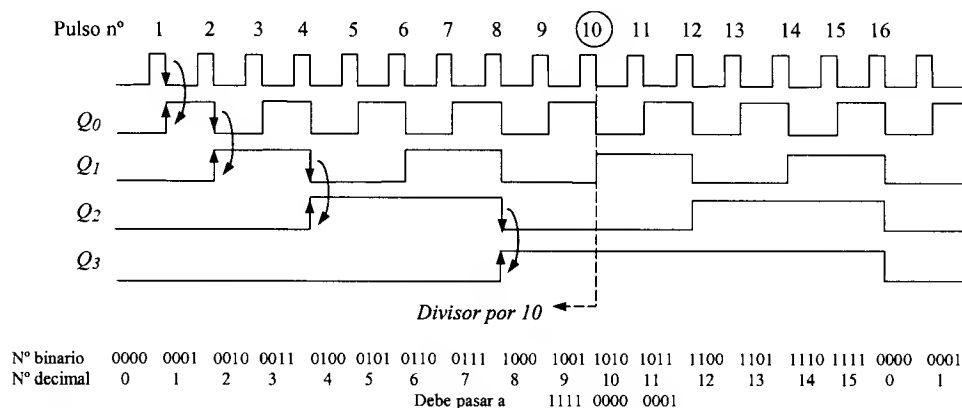
Obsérvese de nuevo que la secuencia a contar no tiene por que estar producida por un reloj o un oscilador que constantemente genera pulsos. El carácter asíncrono de este contador permite que en cada momento ocurran las transiciones que dictan las entradas presentes en ese momento. Si durante cierto intervalo de tiempo no llegan pulsos nuevos el primer biestable no vuelca y, consecuentemente, tampoco vuelcan los otros. Además, el vuelco lo controla  $Q$  ó  $\overline{Q}$  dependiendo del valor de  $x$  en ese momento, ( $x=1$  incrementa,  $x=0$  decrementa).

Hasta ahora sólo hemos usado las entradas  $J$ ,  $K$  y  $Ck$  de los biestables, pero las posibilidades de diseño se amplían y simplifican si hacemos uso de las entradas “*preset*” ( $Pr$ ) y “*clear*” ( $Cl$ ) que son de naturaleza asíncrona y dominan al resto, tal como hemos visto en el capítulo anterior al estudiar los biestables J-K.

Así por ejemplo, a partir de un contador asíncrono binario de  $N$  bits ( $2^N$  estados), podemos obtener divisores por cualquier número  $Q$  (menor y distinto de  $2^N$ ) usando el *preset* y/o el *clear* para cortar la secuencia y volver al estado inicial en el momento que deseemos. Basta detectar la configuración predecesora y usarla para disparar las entradas de “*puesta a cero*” (*clear*) y “*puesta a uno*” (*preset*). Este problema es representativo de un tipo general de circuitos divisores de frecuencia en los que a partir de un tren de impulsos del reloj principal (el de más alta frecuencia) se nos pide otro tren de pulsos de frecuencia más baja de forma que cada  $Q$  pulsos del reloj de entrada se produzca un pulso a la salida del divisor. Lo importante es tener claro inicialmente lo que se entiende

por un divisor por  $Q$ . Es un circuito que produce un cambio en su salida (salto de nivel o pulso) cuando han ocurrido  $Q$  bajadas en la entrada de pulsos del primer biestable. Otra aplicación importante de estas entradas de *preset* y *clear* es facilitar la carga de cualquier estado inicial en el contador. Recordemos la tabla de transición de estados del J-K y veamos estas dos aplicaciones.

Supongamos, para concretar, que queremos modificar un contador binario de 4 bits para obtener un divisor por 10, primero usando el *preset* y después usando el *clear*. La figura 9.22 muestra el cronograma marcado para la solución con *preset*.

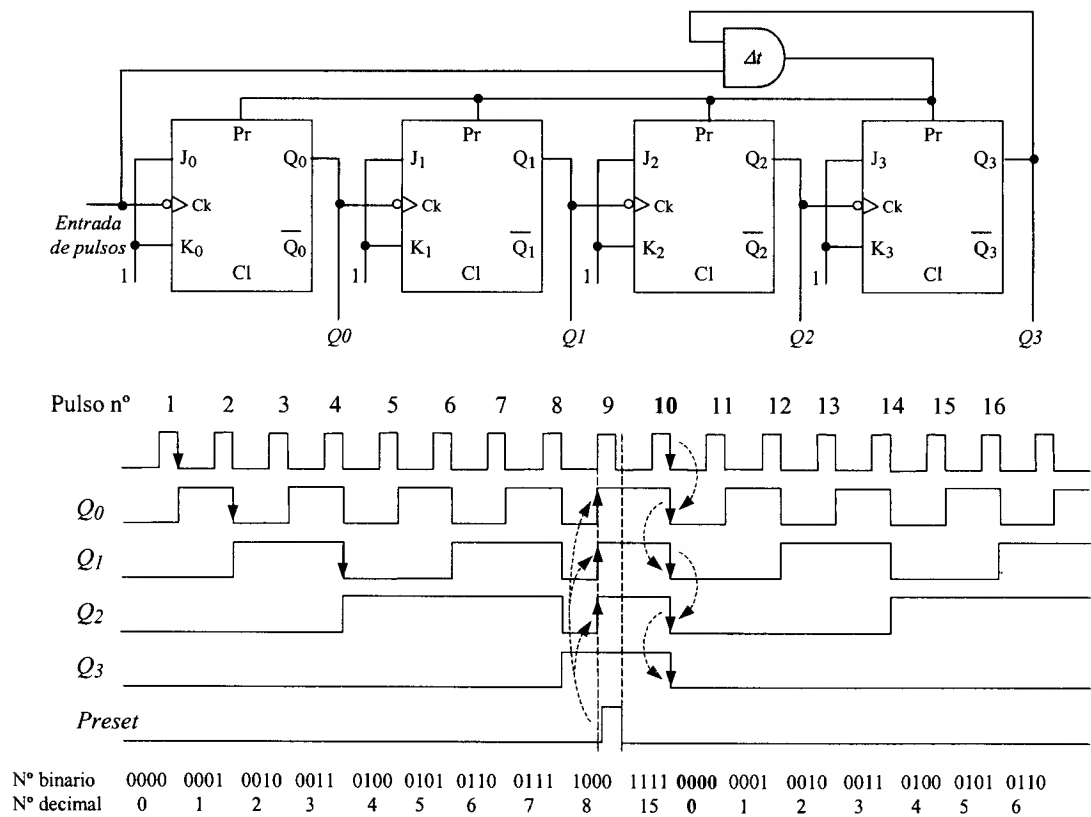


**Figura 9.22.** Cronograma del contador de 4 bits en el que se ha marcado el pulso número 10 con el fin de usarlo para diseñar un divisor por 10.

Como puede observarse en este cronograma, para realizar un divisor por 10 debemos detectar el pulso número 9 que corresponde a la configuración del contador 1000 y así generar el pulso de *preset* que actuará durante el pulso número 10, poniendo a 1 a todos los biestables para que con el siguiente pulso de entrada (pulso número 11) empiece de nuevo el contador a contar desde 0000. Así, el circuito adicional tendrá que detectar que  $Q_3$  está en alta y pasar a alta la salida de  $Q_0$ ,  $Q_1$  y  $Q_2$  durante el siguiente pulso de entrada (pulso número 10). Para ello basta con pasar a uno  $Q_0$ ,  $Q_1$  y  $Q_2$  puesto que  $Q_3$  ya lo está. Sin embargo, para que se repita cíclicamente, debemos asegurar que  $Q_0Q_1Q_2Q_3=1111$  de forma que, con el siguiente pulso de entrada, el contador pase a  $Q_0Q_1Q_2Q_3=0000$  y vuelva a empezar. Así, ponemos todos los biestables a uno mediante la entrada de *preset*. Por tanto, el *preset* deberá activarse cuando se cumplan las dos condiciones: (a) que haya un pulso de entrada y (b) que  $Q_3=1$ . Si se supone el *preset* activo en alta, bastará una puerta AND como



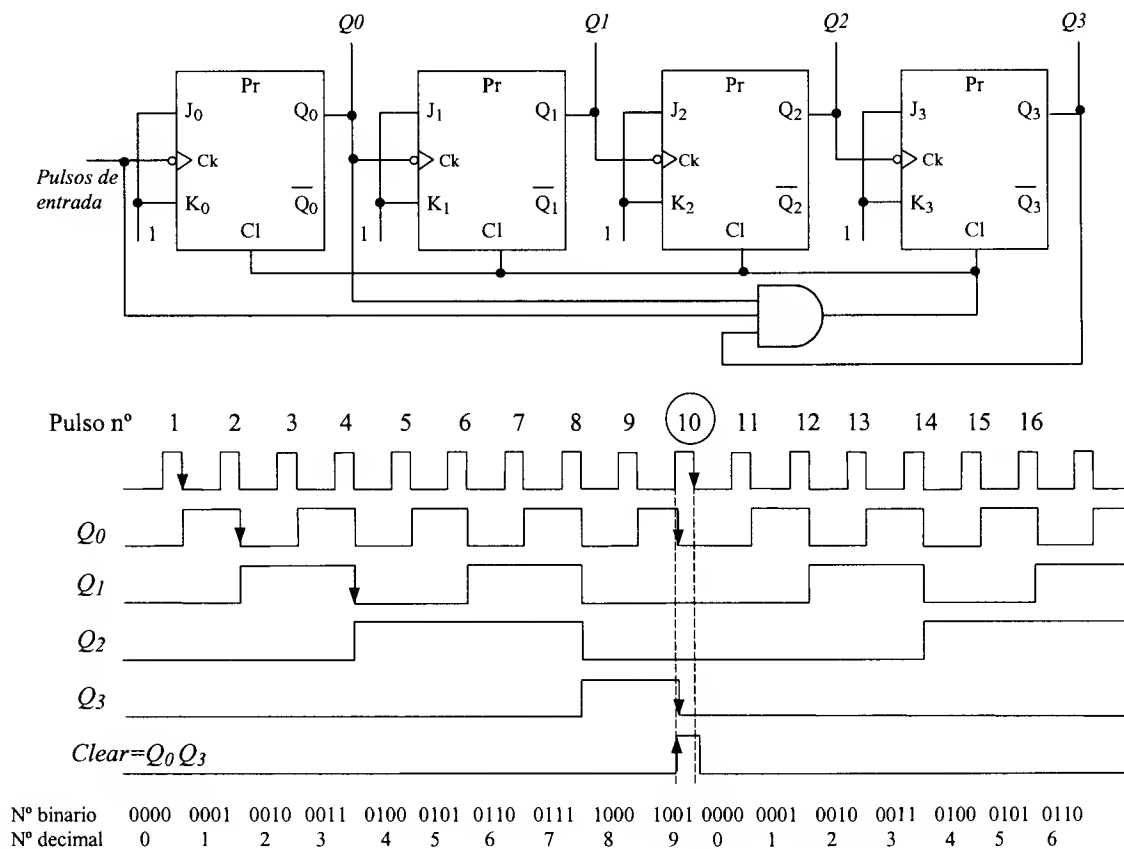
muestra la *figura 9.23*. El pulso de preset generado a la salida de la puerta AND actúa sobre todos los biestables, forzándolos a que pasen a "1". Así, los biestables permanecen en este estado hasta la bajada del siguiente pulso de entrada (el número 10) que hará que pasen automáticamente a "0", empezando a contar de nuevo, tal como ya habíamos mencionado. Es decir, cada 10 pulsos de entrada el contador empieza a contar de nuevo. La *figura 9.23* muestra el circuito completo del divisor por 10 y su cronograma correspondiente, haciendo énfasis en las conexiones causales entre las distintas transiciones de nivel.



**Figura 9.23.** Circuito y cronograma de un divisor por 10 del número de pulsos de entrada, usando un contador asíncrono y controlando el final del ciclo a través del preset .

Normalmente en los circuitos integrados comerciales las entradas de *preset* y *clear* son activas en baja, pero bastaría con cambiar la puerta AND por una NAND.

Podíamos haber usado el *clear* para llevar a 0 a todos los biestables en el pulso en el que queremos que se reinicie el contador. Así en este caso de divisor por 10, podemos en vez de pasar de la configuración 1000 (8) a la 1111 (15) para que el pulso número 10 pase al contador a la configuración 0000, lo podemos hacer detectando que ha llegado el pulso número 10, es decir, que están los biestables en el estado 1001 (9) y a continuación actuar sobre el *clear* de los biestables cuyo siguiente estado no sea de forma natural un 0, para pasarlos a 0. En este caso se detecta mediante una puerta AND a la que le entra  $Q_0$  y  $Q_3$  y la salida de esta puerta se conecta al *clear* de los biestables (aunque bastaría con conectarla sólo al *clear* de  $Q_3$  y  $Q_1$  que son los únicos biestables cuyo siguiente estado sería 1 y que hemos de pasar a 0). La *figura 9.24* muestra el circuito y el cronograma correspondiente.



**Figura 9.24.** Circuito y cronograma de un divisor por 10 controlado mediante el clear.

### 9.5.2. Contadores Síncronos

Los contadores asíncronos presentan una serie de dificultades entre las que cabe destacar:

1. La separación entre impulsos a contar y, por consiguiente, la frecuencia máxima de trabajo del contador, depende de la suma de los retardos que introducen los sucesivos biestables. Este retardo es acumulativo, al irse disparando cada biestable sólo cuando se ha realizado una transición en el biestable anterior.
2. Los estados estables del contador no se alcanzan siempre en el mismo tiempo. Este depende del número de biestables que deban realizar transiciones para alcanzar cada estado concreto.

La solución a gran parte de estos problemas se consigue con los contadores síncronos en los que todos los biestables que tienen que cambiar lo hacen de forma prácticamente simultánea. El precio que hay que pagar es la mayor complejidad en la lógica de interconexión.

En los contadores síncronos, la señal de reloj entra a todos los biestables y las entradas  $J$  y  $K$ , en vez de quedar unidas a un nivel constante ( $J = K = 1$ ), se usan para controlar las transiciones de estado en cada uno de los biestables de acuerdo con la secuencia que se quiere conseguir, tal como vimos en los dos primeros apartados de este tema.

Es decir, en contadores síncronos se usan los procedimientos generales del diseño secuencial calculando las funciones de excitación de cada biestable a partir de la secuencia de estados por la que se desea que pase el contador. Veámoslo con un ejemplo.

Supongamos que queremos convertir en síncrono el contador asíncrono de 3 bits de la *figura 9.20*. Para ello conectamos la señal externa de los pulsos del reloj a la entrada  $Ck$  de todos los biestables para que todas las conmutaciones se realicen simultáneamente y después calculamos las funciones de excitación ( $J_2, K_2; J_1, K_1; J_0, K_0$ ) necesarias para que cada uno de los biestables conmute de acuerdo con la secuencia de estados. Es decir, para que 000 pase a 001, basta con que conmute el biestable  $Q_0$ , para que el siguiente estado sea el 010, deberán conmutar  $Q_0$  (de “1” a “0”) y  $Q_1$  (de “0” a “1”) y así sucesivamente.

Traemos aquí por comodidad la tabla de síntesis con biestables J-K. Es decir, los valores que deben tener  $J$  y  $K$  para que su estado cambie de  $Q_n$  a  $Q_{n+1}$  en cada una de las 4 formas posibles.

$Q_n$	$Q_{n+1}$	$J (set)$	$K (reset)$
0	→ 0	0	*
0	→ 1	1	*
1	→ 0	*	1
1	→ 1	*	0

$$J = \overline{Q_n} Q_{n+1}$$

$$K = Q_n \overline{Q_{n+1}}$$

[9.50]

Cuando se parte de 0, manda  $J$  y el valor necesario en  $J$  es el que se desea en  $Q_{n+1}$ , y no importa el valor de  $K$ . Inversamente, cuando el estado inicial del biestable es 1, manda  $K$  de forma que para que  $Q_n$  pase de “1” a “0”,  $K$  debe ser “1” y para que permanezca en “1”,  $K$  debe ser 0. En estos casos no importa el valor de  $J$ . Al construir la tabla hemos matizado las celdas de la tabla y reforzado con negritas todos los bit que sufren cambios.

La figura 9.25 resume estas transiciones y los valores de las variables de control necesarios para que se produzcan.

	$Q_2^n$	$Q_1^n$	$Q_0^n$	$Q_2^{n+1}$	$Q_1^{n+1}$	$Q_0^{n+1}$	$J_2$	$K_2$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	0	0	0	1	0	*	0	*	1	*
1	0	0	1	0	1	0	0	*	1	*	*	1
2	0	1	0	0	1	1	0	*	*	0	1	*
3	0	1	1	1	0	0	1	*	*	1	*	1
4	1	0	0	1	0	1	*	0	0	*	1	*
5	1	0	1	1	1	0	*	0	1	*	*	1
6	1	1	0	1	1	1	*	0	*	0	1	*
7	1	1	1	0	0	0	*	1	*	1	*	1

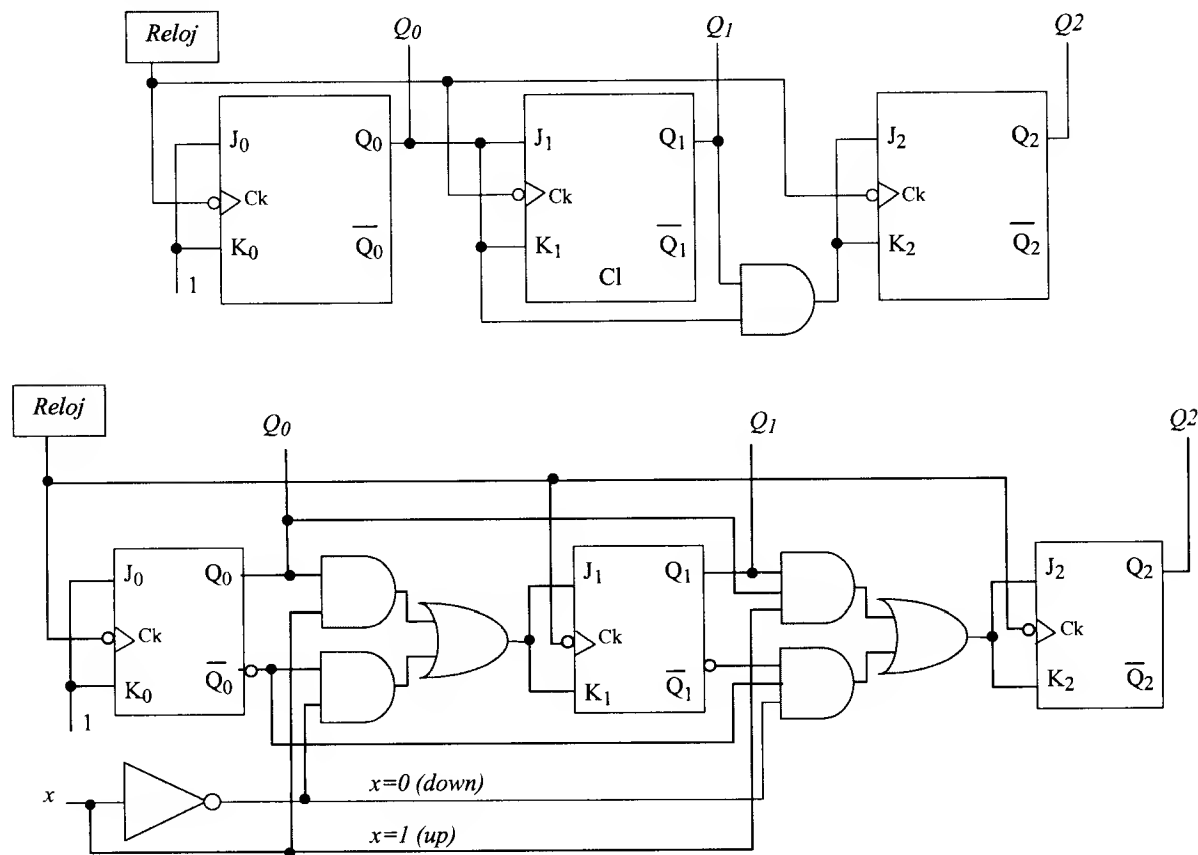
**Figura 9.25.** Transiciones y señales de control necesarias para producirlas en un contador síncrono de 8 estados diseñado con 3 biestables J-K.

Ahora, por simple inspección de la tabla de la figura 9.25 ,y recordando que los asteriscos pueden tomarse a conveniencia como ceros o como unos, podemos escribir;

$$J_0 = K_0 = 1 \quad J_1 = K_1 = Q_0 \quad J_2 = K_2 = Q_1 Q_0 \quad [9.51]$$

Es decir, todos los biestables poseen una entrada directa del reloj y cada uno de ellos es

excitado por el producto de los estados de salida de los anteriores. La *figura 9.26.a* muestra el circuito correspondiente.



**Figura 9.26.** (a) Contador síncrono con acarreo paralelo. (b) Conversión en reversible usando un MUX 2 a 1 y ambas salidas ( $Q$  y  $\bar{Q}$ ).

De forma análoga a la expuesta en el caso asíncrono, es fácil pasar de un contador *síncrono ascendente* a uno *reversible* (ascendente/descendente), usando un multiplexo 2 a 1 entre cada dos biestables para poder seleccionar si la excitación procede de la salida  $Q$  (cuenta ascendente) o de la  $\bar{Q}$  (cuenta descendente). El control del multiplexo lo realiza una señal externa  $x$  tal que para  $x = 1$  cuenta hacia arriba y para  $x = 0$ , decrece la cuenta. Por consiguiente, las ecuaciones de disparo serán:

$$J_0 = K_0 = 1$$

$$J_1 = K_1 = Q_0 \cdot x + \overline{Q_0} \cdot \overline{x}$$

$$J_2 = K_2 = Q_1 \cdot Q_0 \cdot x + \overline{Q_1} \cdot \overline{Q_0} \cdot \overline{x}$$

[9.52]

Si ahora añadimos la entrada de reloj, obtenemos el circuito de la *figura 9.26.b*

### 9.5.3. Aplicación del Método General a la Síntesis de Contadores con PLDs

Ahora que hemos visto la síntesis de contadores asíncronos y síncronos pensando en cada caso en la función específica que se tenía que realizar y en la forma de conectar los biestables para alcanzarla, vamos a describir la síntesis de contadores usando el procedimiento general descrito en el apartado 9.2 de este tema. Así, no tenemos que preocuparnos de si queremos que cuente por 5 o por 7 ó si debe o no ser reversible. Un contador de  $N$  bits es un autómata finito de  $2^N$  estados descritos por su diagrama de transición de estados y, como tal, puede sintetizarse con PLDs de  $N$  macroceldas de salida que incluyen, al menos, un biestable D cada una. Vamos a verlo sobre un ejemplo concreto de contador reversible de 3 bits (8 estados).

#### P.1. Descripción de la función

Un contador reversible síncrono de 8 estados es un circuito secuencial en el que existe una entrada de control (sea  $x$ ) tal que cuando esta entrada está en alta ( $x=1$ ), el contador incrementa su contenido con cada pulso del reloj, pasando desde (000) hasta (111). Inversamente, cuando la entrada de control está en baja ( $x=0$ ), cada pulso de reloj decrementa en una unidad el contenido del contador (es decir, su *estado*).

#### P.2. Representación

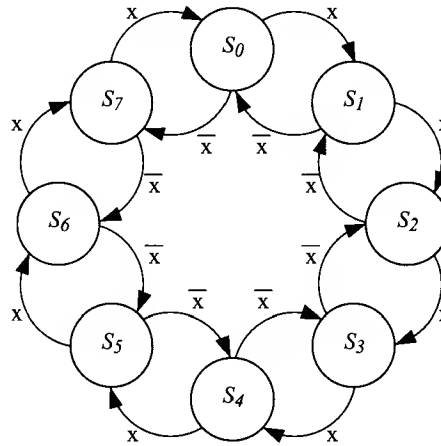
La *figura 9.27* representa este comportamiento en forma de diagrama de transición de estados. La *figura 9.28* muestra las dos matrices de transición de estados  $(T_{ij}^0, T_{ij}^1)$  asociadas a cada una de las dos configuraciones de entrada posibles ( $x=0$  y  $x=1$ ) y la *figura 9.28* resume las dos anteriores mostrando la matriz funcional correspondiente, obtenida como suma de los productos de las entradas por sus matrices de transición asociadas:

$$M(x) = T^0 \cdot \overline{x} + T^1 \cdot x$$

[9.53]

### P.3, P.4 y P.5. Asignación de estados.

Los pasos P.3 y P.4 son obvios aquí. Primero, no hay estados redundantes. Para contar hasta  $n$  hacen falta  $N$  biestables, siendo  $n \leq 2^N$ . En nuestro caso hemos elegido 8 estados para optimizar el diseño con tres biestables ( $2^3 = 8$ ). Al hacer el diseño con retardos (D), el paso P.4 se simplifica también y nos queda P.5, que es la asignación de estados. El procedimiento seguido es también sencillo y ha sido ilustrado previamente. Se muestra en la *figura 9.28*. Los tres biestables ( $D_2, D_1, D_0$ ) han sido elegidos de forma que sus salidas sean:  $Q_0 = \text{bit menos significativo}$ ,  $Q_1 = \text{bit intermedio}$  y  $Q_2 = \text{bit más significativo}$ .



**Figura 9.27.** Diagrama de transición de estados de un contador reversible de tres bits.

### P.6. Síntesis de $D_0, D_1$ y $D_2$ .

Por simple inspección de la matriz funcional de la *figura 9.28*, vemos que  $Q_2$  está en alta en los estados  $S_4, S_5, S_6$  y  $S_7$ , luego para obtener la expresión de  $D_2(t)$  sumaremos los productos de las entradas por los estados iniciales en las que se consigue ese estado final. Es decir:

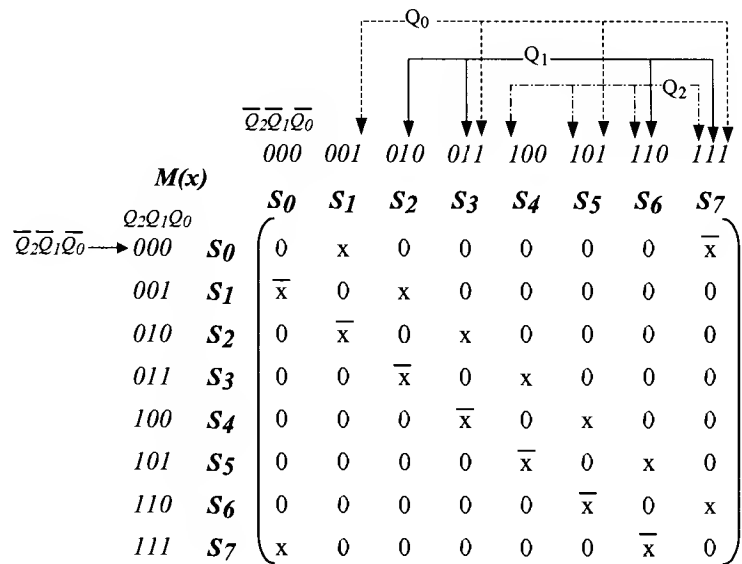
$$\begin{aligned}
 D_2(t) = & \bar{Q}_2(t - \Delta t) \cdot \bar{Q}_1(t - \Delta t) \cdot \bar{Q}_0(t - \Delta t) \cdot \bar{x}(t - \Delta t) + \bar{Q}_2 Q_1 Q_0 x + \\
 & + Q_2 \bar{Q}_1 \bar{Q}_0 x + Q_2 \bar{Q}_1 Q_0 (\bar{x} + x) + Q_2 Q_1 \bar{Q}_0 (\bar{x} + x) + Q_2 Q_1 Q_0 \bar{x}
 \end{aligned}
 \tag{9.54}$$

Por facilidad de escritura no ponemos el  $(t - \Delta t)$  en todos los términos mínimos que hay a la derecha del igual. Sólo lo hemos puesto en el primer término mínimo.

$T^0(x=0)$		$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$
		000	001	010	011	100	101	110	111
$S_0$	000	0	0	0	0	0	0	0	1
$S_1$	001	1	0	0	0	0	0	0	0
$S_2$	010	0	1	0	0	0	0	0	0
$S_3$	011	0	0	1	0	0	0	0	0
$S_4$	100	0	0	0	1	0	0	0	0
$S_5$	101	0	0	0	0	1	0	0	0
$S_6$	110	0	0	0	0	0	1	0	0
$S_7$	111	0	0	0	0	0	0	1	0

$T^1(x=1)$		$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$
		000	001	010	011	100	101	110	111
$S_0$	000	0	1	0	0	0	0	0	0
$S_1$	001	0	0	1	0	0	0	0	0
$S_2$	010	0	0	0	1	0	0	0	0
$S_3$	011	0	0	0	0	1	0	0	0
$S_4$	100	0	0	0	0	0	1	0	0
$S_5$	101	0	0	0	0	0	0	1	0
$S_6$	110	0	0	0	0	0	0	0	1
$S_7$	111	1	0	0	0	0	0	0	0

**Figura 9.28.** Matrices de transición de estados para los dos valores de la entrada ( $x = 0$ ,  $x = 1$ ) en un contador reversible.



**Figura 9. 29.** Matriz funcional correspondiente al contador de la figura 9.27. Recordemos que se obtiene multiplicando cada valor de la configuración de entrada  $(\overline{x}, x)$  por la matriz de transición correspondiente y sumando ambos productos ya que las configuraciones de entrada son mutuamente exclusivas. Por facilitar el análisis se ha repetido al comienzo de fila y columna la codificación del estado. También se incluyen las flechas que agrupan el conjunto de estados *finales* en los que un determinado biestable está en alta.

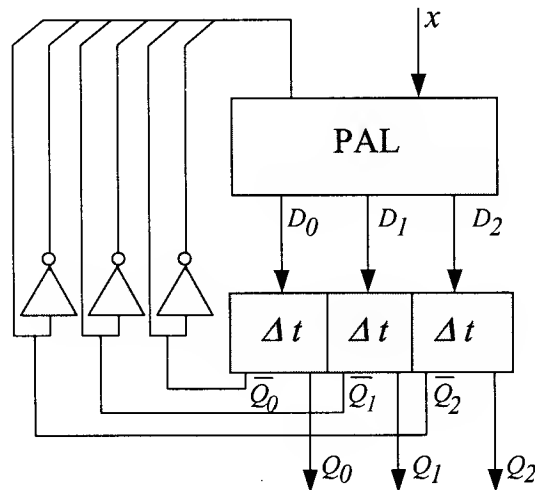


Repitiendo el mismo procedimiento podemos obtener las expresiones correspondientes para las señales de excitación de los otros dos biestables,  $D_1(t)$  y  $D_0(t)$ :

$$D_1(t) = \bar{Q}_2 \bar{Q}_1 \bar{Q}_0 \bar{x} + \bar{Q}_2 \bar{Q}_1 Q_0 x + \bar{Q}_2 Q_1 \bar{Q}_0 x + \bar{Q}_2 Q_1 Q_0 \bar{x} + Q_2 \bar{Q}_1 \bar{Q}_0 \bar{x} + Q_2 \bar{Q}_1 Q_0 x + Q_2 Q_1 \bar{Q}_0 x + Q_2 Q_1 Q_0 \bar{x} \quad [9.55]$$

$$D_0(t) = \bar{Q}_2 \bar{Q}_1 \bar{Q}_0 (x + \bar{x}) + \bar{Q}_2 Q_1 \bar{Q}_0 (x + \bar{x}) + Q_2 \bar{Q}_1 \bar{Q}_0 (x + \bar{x}) + Q_2 Q_1 \bar{Q}_0 (\bar{x} + x) \quad [9.56]$$

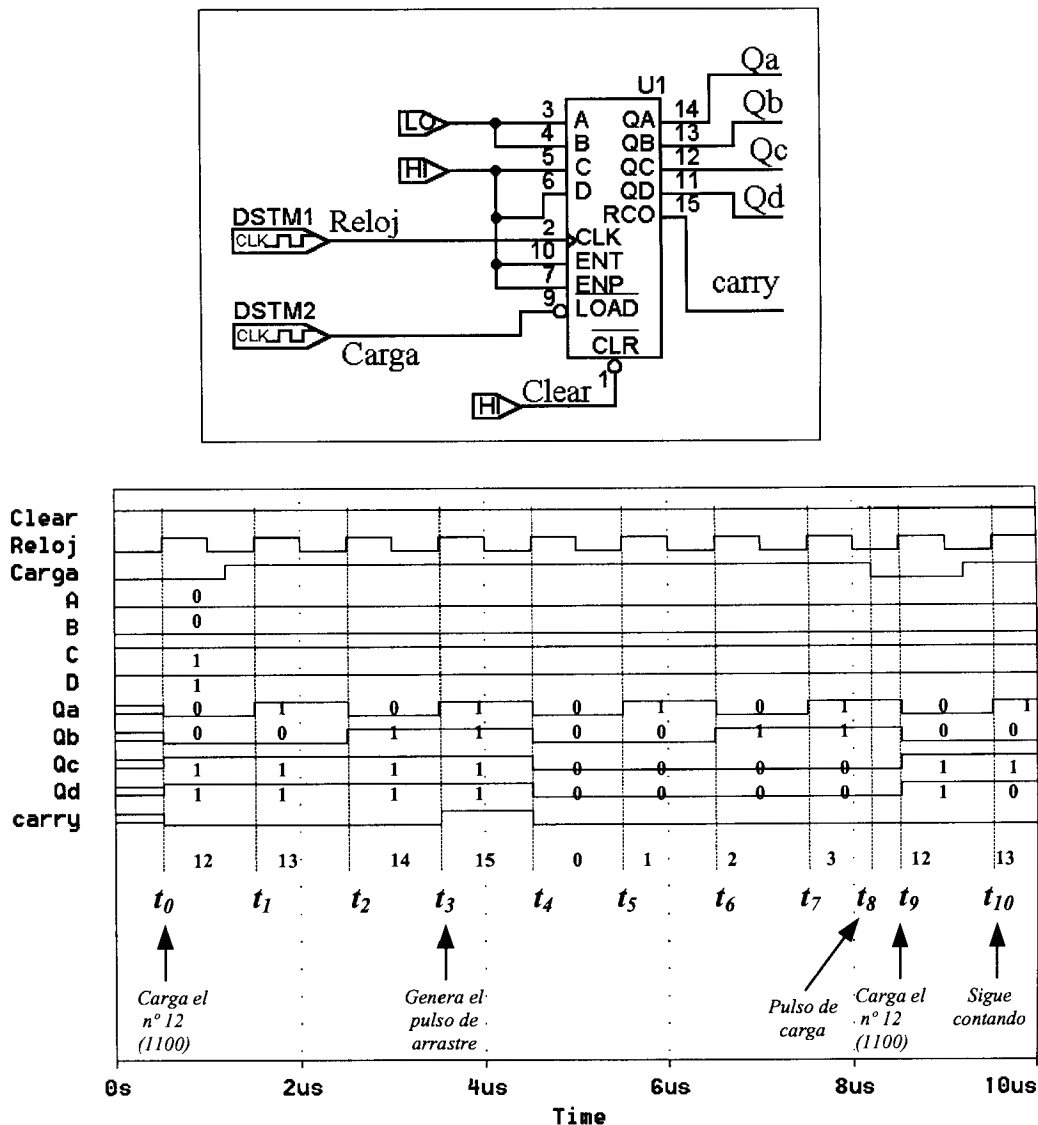
Para sintetizar este contador usando lógica programable necesitamos una PAL de cuatro entradas (la externa,  $x$ , y las realimentaciones del estado,  $Q_2$ ,  $Q_1$  y  $Q_0$ ) y tres salidas. La parte combinacional de la PAL sintetiza las tres funciones de excitación ( $D_2$ ,  $D_1$  y  $D_0$ ) y sus biestables de salida almacenan el estado. Un circuito tipo PAL16R8 posee más capacidad computacional de la necesaria. La *figura 9.30* muestra de forma cualitativa la conexión necesaria para este contador reversible sin bajar al detalle de los términos mínimos. La flexibilidad de la estructura hace que podamos modificar el tipo de contador, decidiendo que actúe como divisor por 7, por 6, ó por 5, por ejemplo.



**Figura 9.30.** Contador reversible de ocho estados sintetizado mediante una PAL tipo 16R8 que ya incluye los biestables D. Ver tema 7 para precisar los detalles de la programación de los dos multiplexos de las macroceldas de salida.

### 9.5.4. Simulación y ejemplos de Contadores

Vamos a desarrollar un ejemplo de uso del simulador para estudiar el comportamiento de los contadores y después haremos mención a algunas de las soluciones integradas en MSI. La *figura 9.31* muestra el esquema de conexión y un cronograma correspondientes al contador síncrono SN74163.



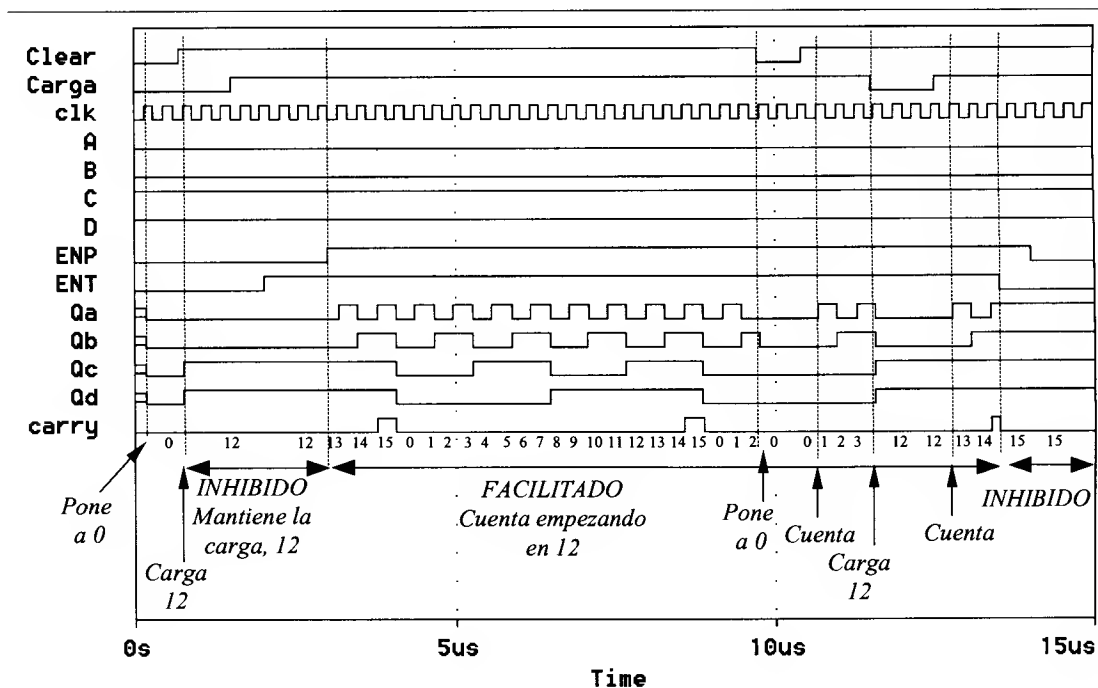
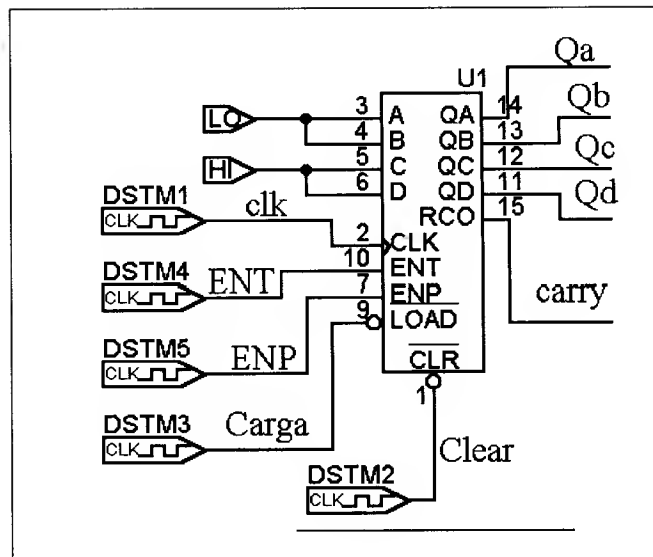
**Figura 9.31.** Esquema de simulación del contador síncrono SN74163 junto con el cronograma correspondiente

Primero vamos a simular el contador con las entradas de *CLEAR*, *ENT* y *ENP* inactivas (en alta). Con ello el contador está en modo de habilitado. Para la señal de *LOAD* (carga) usaremos un generador de pulsos, de forma que cuando esta señal de carga está en baja y el pulso de reloj pasa a alta ( $t_0$ ) el contador se carga con el valor que tenemos puesto en las entradas, en el caso del ejemplo tenemos  $A=B=0$  y  $C=D=1$ , es decir, queremos cargar el número 12 en decimal (1100 en binario, A el LSB y D es el MSB). A partir de este instante  $t_0$ , como es un contador síncrono (sincronizado con los flancos de subida) cada vez que el reloj pasa de baja a alta el contador adelanta un paso en la cuenta. Como lo hemos cargado con el número 12 en  $t_1$  pasará a 13 (1101) a continuación en  $t_2$  pasará a 14, en  $t_3$  pasará a 15.

Obsérvese cómo en el instante  $t_3$ , en el que aparece el número 15 en la salida de arrastre (*RCO*), se genera un pulso avisándonos de que el contador ha pasado por el máximo número por el que puede pasar puesto que es de 4 bits (sólo tiene 16 estados). La función de esta señal es la misma que el *carry* de un sumador. Es decir, la función de esta señal de control es permitir la conexión en cascada de varios contadores para aquellos casos en los que es necesario contar un número de pulsos superior a los 16 estados de este circuito.

Siguiendo con la descripción del contador, en el siguiente intervalo de tiempo,  $t_4$ , el contador pasará a la configuración 0000, y seguirá contando, si no se modifican ninguna de las señales de control. De nuevo, en el instante  $t_8$ , aparece otro pulso de carga y como el contador está en el estado 0011 (3 en decimal), en la siguiente subida del reloj,  $t_9$ , el contador cambia de estado. Lo usual sería que el contador pasara al estado siguiente (0100, 4 en decimal) pero como ha llegado un pulso de carga, vuelve a cargarse con el valor de la entrada 1100 (12 en decimal). A partir de aquí seguirá contando de igual forma que lo hizo antes, como si estuviera en el instante  $t_0$ .

Veamos ahora cómo actúa la señal asíncrona de puesta a cero (*clear*) y las señales de inhibición/facilitación (*ENT* y *ENP*) que hasta ahora las habíamos mantenido en alta para que el contador estuviera siempre habilitado. Como siempre, para ver como actúan estas señales introducimos generadores de pulsos. En el caso de *ENT* y *ENP* hemos combinado dos frecuencias para que aparezcan de forma sucesiva las cuatro configuraciones lógicas posibles. El circuito y el cronograma correspondientes son los que se muestran en la *figura 9.32*, donde para facilitar la comprensión hemos introducido en la parte inferior del cronograma las descripciones de los distintos comportamientos del contador correspondientes a cada segmento temporal.



**Figura 9.32.** Estudio del efecto de las señales clear, ENT y ENP en el comportamiento del contador SN74163

El proceso comienza con la puesta a cero, después se carga el número 12 , a continuación hay una serie de pulsos de reloj durante los cuales el contador permanece inhibido manteniendo el estado (el número 12 que se le cargó anteriormente) después hay un intervalo temporal durante el cual el contador está facilitado por lo que sigue contando los pulsos del reloj a partir de 12 y volviendo a cero cada vez que llega a 15. Iniciamos una nueva fase poniendo a cero al contador, haciéndole contar, cargándole de nuevo 12, contando de nuevo, entrando de nuevo en fase de inhibición, etc...

En el cronograma se observa cómo la señal de *clear* controla la puesta a cero y cómo cuando las dos señales *ENT*, *ENP* están en alta el contador está facilitado para contar mientras que en sus otras tres configuraciones (00, 01, 10) el contador está inhibido. Finalmente, cuando estas dos señales está en baja permiten la puesta a cero del contador así como la carga del valor inicial que está presente en la entrada.

Existen soluciones integradas para contadores asíncronos y síncronos, disparados por flancos en ambos casos, en general con posibilidad de carga en paralelo. Los asíncronos se clasifican a su vez en términos del estado en el que finalizan la cuenta. Tenemos así, contadores de décadas (74196), binarios de 4 bits (74197), divisores por 12 (7492), etc.

Los síncronos se clasifican también en términos del estado final al que llegan contando, y dependiendo de si son o no reversibles y de si admiten o no carga paralelo, síncrona o asíncrona. Así, el 74160 es un contador de décadas con carga paralela síncrona. El 74190 es un contador de décadas, reversible y con posibilidad de carga paralelo asíncrona y el 74191 es un contador binario reversible, con *preset* y *clear* (con posibilidad de carga paralelo asíncrona)<sup>(\*)</sup>.

El objetivo de este apartado ha sido mencionar una parte del repertorio de circuitos contadores que ofrecen las casa comerciales. A partir de ahora es poco probable que nos encontremos con la necesidad de diseñar un contador usando puertas y biestables. Más bien el problema frecuente será tener que usar estos circuitos contadores como módulos intermedios en el diseño de otras funciones secuenciales de mayor complejidad. Por ejemplo, diseñar unidades de control de cualquier sistema de

---

<sup>(\*)</sup> Como siempre, conviene visitar en la red la información que suministran las distintas casa comerciales para conocer el estado actual de los circuitos, sus hojas de datos y las notas de aplicación correspondientes. Por ejemplo [www.fairchildsemi.com](http://www.fairchildsemi.com), [www.onsemi.com](http://www.onsemi.com).

cálculo, para direccionar una memoria, para producir secuencias de trenes de impulsos previamente especificadas, etc.

## 9.6. REGISTROS DE DESPLAZAMIENTO

Un registro de desplazamiento de  $N$  bits es un circuito secuencial que posee la estructura de la *figura 9.31*. Consta de  $N$  biestables D conectados en cascada, un reloj y distintas facilidades de carga y descarga en serie y en paralelo. En su esquema básico, la información entra por el primer biestable y es transferida a los siguientes ante sucesivos pulsos de reloj de forma que de cada biestable D pasa al siguiente su contenido. Finalmente, el contenido del último biestable se pierde, o entra a realimentación.

El contenido del primer biestable se obtiene en general mediante una función lógica de la entrada externa,  $x$ , y del estado de todos los biestables ( $Q_0, Q_1, \dots, Q_{N-1}$ ) de forma que

$$\begin{aligned} D_{N-1}(t) &= Q_{N-2}(t - \Delta t) \\ D_{N-2}(t) &= Q_{N-3}(t - \Delta t) \end{aligned}$$

$$D_0(t) = f[x(t - \Delta t); Q_0(t - \Delta t); Q_1(t - \Delta t); \dots Q_{N-1}(t - \Delta t)]$$

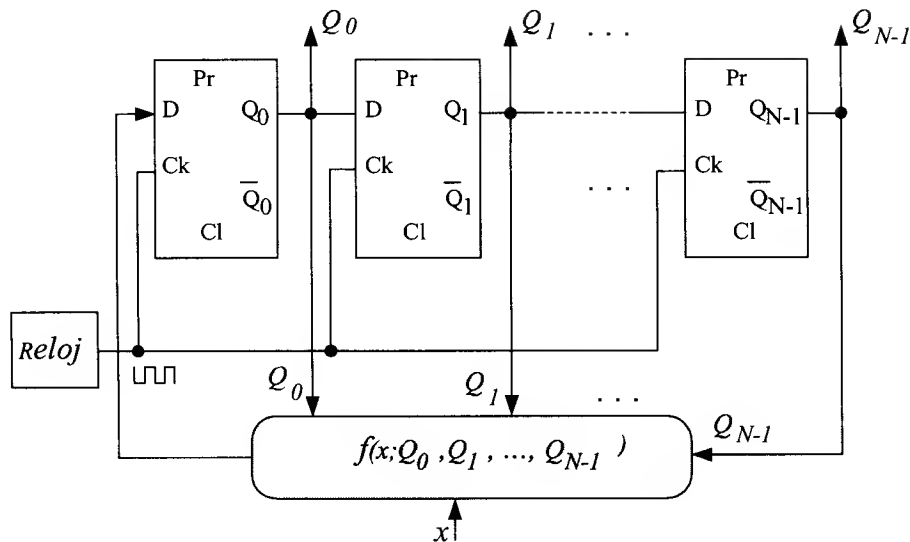
[9.57]

Como se observa en la *figura 9.33*, los registros de desplazamiento son un caso particular de circuito secuencial en el que se fuerza la conectividad de todos los biestables menos la del primero que admite datos externos directamente o a través de una función de realimentación. En las aplicaciones más frecuentes, esta función de realimentación no existe o se limita a dejar circular los datos, de forma que el registro de desplazamiento actúa como una memoria.

Las facilidades adicionales que incluyen las soluciones integradas permiten clasificar a los registros de acuerdo con la forma en que reciben y entregan los datos. Así tenemos:

1. ( $S - S$ ), entrada serie, salida serie.
2. ( $S - P$ ), entrada serie, salida paralelo.
3. ( $P - S$ ), entrada paralelo, salida serie.

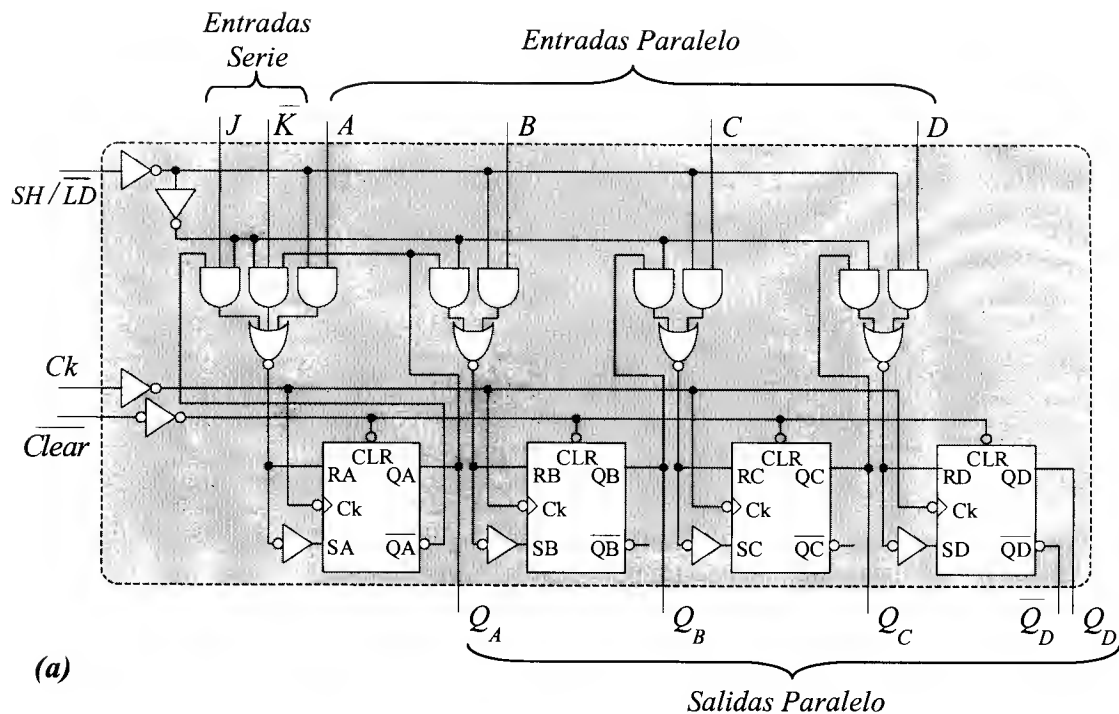
4.  $(P - P)$ , entrada paralelo, salida paralelo.
5.  $(P - P)$ , entrada paralelo, salida paralelo.



**Figura 9.33.** Estructura general de los registros de desplazamiento.

Las principales aplicaciones de los registros de desplazamiento son la conversión de datos de serie a paralelo y viceversa. También se usan como elementos de memoria transitoria, como subsistemas de ayuda a las unidades aritmético-lógicas (desplazar es equivalente a multiplicar o dividir por dos), como líneas de retardo digital, en sistemas de adquisición de datos y como generadores de secuencias necesarias para el control secuencial y concurrente de un conjunto de operaciones de cálculo o transferencia de información. Existen soluciones integradas en MSI y LSI en tecnologías TTL, MOS y CCD (Charge Coupled Devices) con una amplia variedad de funciones. Así, en TTL hay registros de cuatro bits con acceso paralelo (7495), bidireccionales (74194), de 5 bits (7476), de 8 bits (74164) y de 16 bits (74LS673). Por razones pedagógicas hemos seleccionado el circuito SN74AS195 para ilustrar los datos de catálogo en este apartado.

El SN74AS195 es un registro de desplazamiento universal de cuatro bits. Acepta carga en serie y en paralelo y su salida está también accesible en serie y en paralelo. La *figura 9.34* muestra su circuito interno y la tabla de control de función y en la *figura 9.35* se muestra el cronograma de una secuencia típica que incluye el borrado, el desplazamiento y la carga.



(b)

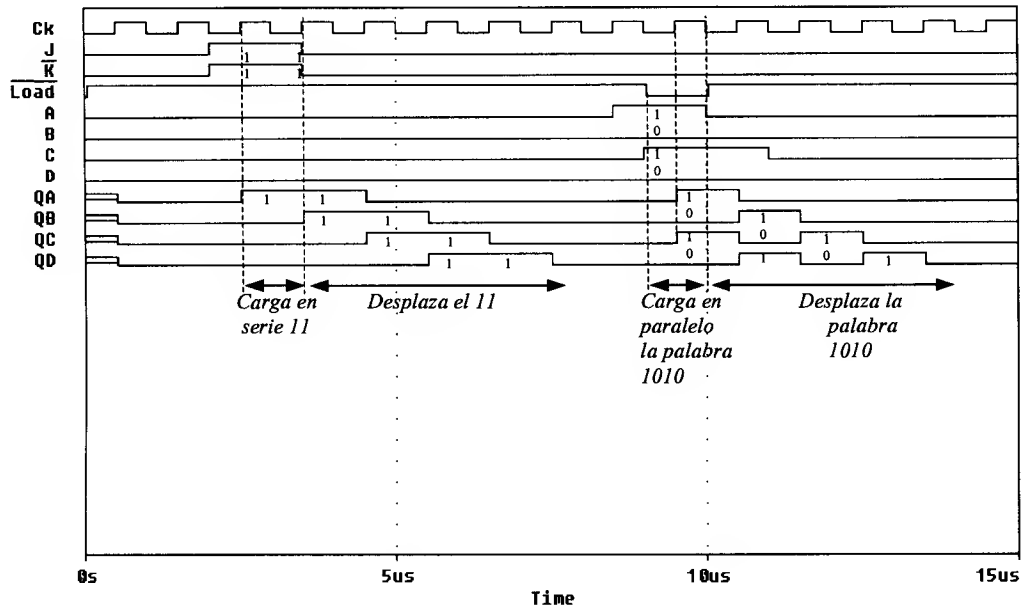
			Entradas							Salidas				
$\overline{Clear}$	$\overline{SH/LD}$	$Ck$	Serie		Paralelo									
			$J$	$\overline{K}$	$A$	$B$	$C$	$D$		$Q_A$	$Q_B$	$Q_C$	$Q_D$	$\overline{Q_D}$
L	x	x	x	x	x	x	x	x		L	L	L	L	H
H	L	$\uparrow$	x	x	a	b	c	d		a	b	c	d	d
H	H	L	x	x	x	x	x	x		$Q_{A0}$	$Q_{B0}$	$Q_{C0}$	$Q_{D0}$	$\overline{Q_{D0}}$
H	H	$\uparrow$	L	H	x	x	x	x		$Q_{A0}$	$Q_{A0}$	$Q_{Bn}$	$Q_{Cn}$	$\overline{Q_{Cn}}$
H	H	$\uparrow$	L	L	x	x	x	x		L	$Q_{An}$	$Q_{Bn}$	$Q_{Cn}$	$\overline{Q_{Cn}}$
H	H	$\uparrow$	H	H	x	x	x	x		H	$Q_{An}$	$Q_{Bn}$	$Q_{Cn}$	$\overline{Q_{Cn}}$
H	H	$\uparrow$	H	L	x	x	x	x		$\overline{Q_{An}}$	$Q_{An}$	$Q_{Bn}$	$Q_{Cn}$	$\overline{Q_{Cn}}$

**Figura 9.34.** Registro de desplazamiento 74195. (a) Circuito interno. (b) Tabla de control de función.

El desplazamiento y la función de carga serie o carga en paralelo están controladas por un multiplexo, como en el caso de los contadores síncronos reversibles. El borrado se realiza



directamente a través de las entradas de *reset* y la función *D* de los biestables se consigue a partir de R-S conectados a través de un inversor. Veamos con más detalle el significado funcional de los distintos terminales.



**Figura 9.35.** Cronograma del registro de desplazamiento 74195, mostrando la evolución temporal de las señales en una secuencia típica de carga serie, desplazamiento, carga paralela y nuevo desplazamiento (la señal de *clear* no se muestra pero durante la carga y el desplazamiento debe permanecer en alta).

Como ya hemos mencionado, el terminal de *clear* sirve para poner a cero todos los biestables. Está activo en baja. El registro es síncrono y los cambios de estado se producen en las subidas del reloj. La otra entrada de control es  $\text{Shift}/\overline{\text{Load}}$  ( $\text{SH}/\overline{\text{LD}}$ ). Cuando está en baja la carga del registro se realiza en paralelo a través de las entradas *A*, *B*, *C* y *D*. Si está en alta, la carga se realiza en serie a través de las entradas *J* y  $\overline{\text{K}}$ .

Para que se inicie el proceso de carga en paralelo, la entrada  $\text{SH}/\overline{\text{LD}}$  debe ponerse a 0 y  $\overline{\text{clear}}$  debe estar en alta (1). Los valores que se quieren cargar deben estar en *A*, *B*, *C* y *D* y entran a la *R* y *S* del biestable correspondiente a través de las puertas NOR de su multiplexo. Al tener  $S = \overline{R}$ , los

biestables actúan como D y transmiten estos valores a sus salidas haciendo  $Q_A = A$ ,  $Q_B = B$ ,  $Q_C = C$  y  $Q_D = D$ .

Como podemos ver en el cronograma de la *figura 9.35*, los biestables pasan a estos valores en el instante en el que, estando  $\overline{SH/LD}$  en 0, el *reloj* pasa a alta. Estos valores se mantienen durante todo un periodo del reloj. Si antes de pasar todo el periodo del reloj, existiera un cambio en  $\overline{SH/LD}$ , manteniéndose en baja las entradas serie ( $J, \overline{K}$ ), cada biestable transferiría su contenido al biestable siguiente, desplazándose la información que hemos cargado. Cada nuevo pulso desplaza una posición la información cargada de forma que tras el cuarto pulso, todos los biestables estarán a 0. Se termina así el proceso de conversión paralelo-serie.

Veamos ahora qué pasa con la carga serie. Se facilita con  $\overline{SH/LD}$  en alta y se controla con ( $J, \overline{K}$ ). Si previamente a la carga serie, mediante *clear*, hemos puesto a cero todas las salidas, con ( $J=1$ , y  $\overline{K}=1$ ), ponemos  $Q_A = 1$ , mientras que  $Q_B = Q_C = Q_D = 0$ , que es el valor que tenían antes. A partir de aquí, si se pasan  $J$  y  $\overline{K}$  a 0, el 1 cargado en  $Q_A$  se irá transfiriendo a los siguientes biestables, en los sucesivos pulsos de reloj.

Si  $J = 1$  y  $\overline{K} = 1$ , la salida  $Q_A$  pasa a uno.

Si  $J = 1$  y  $\overline{K} = 0$ ,  $Q_A(t) = \overline{Q}_A(t - \Delta t)$  y si  $J = 0$  y  $\overline{K} = 1$ , tenemos  $Q_A(t) = Q_A(t - \Delta t)$ . Es decir, la carga serie se realiza a través del biestable A que pasa a 0, pasa a 1, mantiene su estado anterior,  $Q_A(t - \Delta t)$ , ó lo invierte,  $\overline{Q}_A(t - \Delta t)$ , dependiendo de los valores de las variables de control  $J$  y  $\overline{K}$ .

## 9.7. PROBLEMAS

Este capítulo posee dos partes claramente distinguibles:

- a) *Representación, diseño y análisis general con PLD's y biestables D.*
- b) *Diseños específicos de contadores y registros.*

Por consiguiente, los ejercicios que proponemos se encaminan a repasar y afianzar los conocimientos obtenidos en ambas partes. Para aprovechar estos ejercicios es conveniente que se

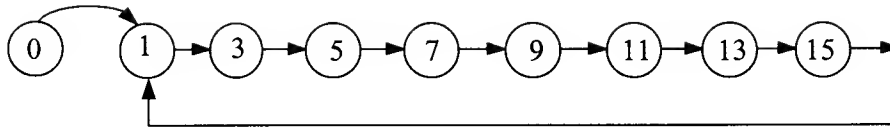
hayamos realizado previamente los correspondientes al tema 8. Allí, tras el estudio de los biestables, propusimos un conjunto de ejercicios relacionados con el análisis y la síntesis de circuitos sencillos, con un único biestable o con dos biestables pero con funciones combinacionales muy sencillas de forma que no fuera necesario conocer procedimientos generales de síntesis y análisis. Aquí queremos hacer énfasis en el procedimiento complementario. Es decir, no necesitamos bajar al detalle de las conexiones para cada ejemplo concreto, porque los procedimientos de síntesis y análisis son generales.

Otro punto importante es el *paralelismo*. En todos los ejercicios del tema 8 y en los correspondientes a contadores y registros, dentro de este tema, la síntesis es serie. La salida de un biestable va al siguiente. En cambio, en el procedimiento general de síntesis con PLD's y biestables D, el circuito calcula en *paralelo*, ya que la salida de todos los biestables, junto con las entradas externas van a la PAL que se encarga de excitar de la forma adecuada a todas las entradas,  $\{D_i\}$ . Así, cualquier cambio de estado conlleva el mínimo retardo. Más exactamente, todos los cambios de estado conllevan el mismo retardo porque no hay ninguna computación global que consuma más tiempo que la computación local que realiza uno cualquiera de los biestables D junto con su función combinacional previa.

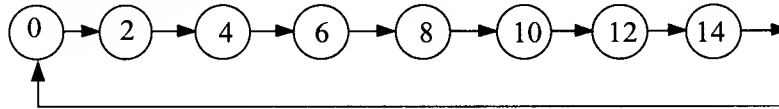
Los ejercicios de la segunda parte (contadores y registros), son distintos. La meta aquí es conseguir que el alumno comprenda cómo funcionan las soluciones integradas correspondientes. Para ello, por ejemplo, se modifica alguna especificación (tal como el estado de vuelta al origen) y se pide que se analice o sintetice el nuevo circuito contador al que da origen esa pequeña modificación en las especificaciones funcionales.

**E.9.1.** *Representar (pasos P.1 y P.2 del procedimiento general), el autómata finito que resuelve el siguiente problema de control de los movimientos de un ascensor en un edificio con planta baja y dos pisos. Considerar como entradas al autómata las órdenes de llamada (ninguna llamada, llamada del piso 0, del piso 1 y del piso 2) y llamadas de dos ó más pisos a la vez) Las salidas del autómata son las órdenes que el controlador digital da a los motores (subir, bajar, quedarse donde está). Finalmente, los estados corresponden a las situaciones distinguibles de ascensor parado en piso 0, 1 ó 2, ascensor subiendo y ascensor bajando. Suponer que cuando el ascensor está en movimiento no acepta órdenes y que cuando, estando parado, recibe dos ó más órdenes sólo atiende a la que le cuesta menos esfuerzo. Es decir, si está en el piso 0 y la llaman del 1 y del 2, irá al primero.*

- a) *Obtener los espacios de entrada y salida y los estados internos necesarios. Asignar las variables lógicas necesarias para su descripción formal.*
  - b) *Sintetizar el autómata resultante usando el procedimiento general con biestables D y PLD's.*
  - c) *Obtener las funciones de salida a partir de las entradas y los estados.*
  - d) *Realizar el proceso inverso de análisis y comprobar que se obtiene la matriz funcional de partida.*
- E.9.2.** *Durante la presentación del material teórico de este tema propusimos la síntesis de un autómata de dos estados con un biestable D (figura 9.14) y el análisis y síntesis de otro autómata de cuatro estados (figuras 9.15 y 9.16). Repetir aquí ambos ejercicios usando biestables J-K, en vez de D.*
- E.9.3.** *Diseñar un autómata universal de dos estados. ¿Cuántas entradas hacen falta para controlarlo?*
- E.9.4.** a) *Diseñar un circuito que detecte la secuencia 111 sin solapamiento, de forma que siempre que vengan tres unos seguidos por la línea de entrada,  $x$ , produzca un uno a la salida, y como respuesta al último uno de la secuencia.*
- b) *Repetir el problema para las subsecuencias 010 y 101.*
- c) *Supongamos ahora que estas dos últimas subsecuencias (010 y 101) vienen por dos líneas separadas. Diseñar un circuito secuencial que compare las entradas por dos líneas separadas ( $x_0$  y  $x_1$ ) y que produzca una salida en alta cuando por  $x_0$  viene 010 y por  $x_1$  viene 101.*
- E.9.5.** *Hemos visto el diseño de contadores asíncronos y el uso del preset para obtener un divisor por 10. Repetir el diseño para conseguir:*
- a) *Un divisor por 12.*
  - b) *Un divisor por 9.*
- E.9.6.** *Diseñar los circuitos que partiendo del estado (0000) y al ritmo de los pulsos del reloj recorra las siguientes secuencias:*
- a) *La secuencia de los números impares*



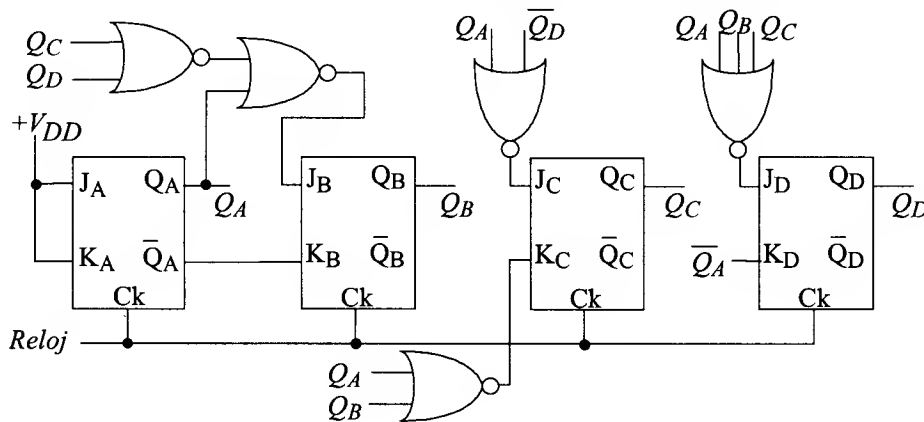
b) La secuencia de los números pares



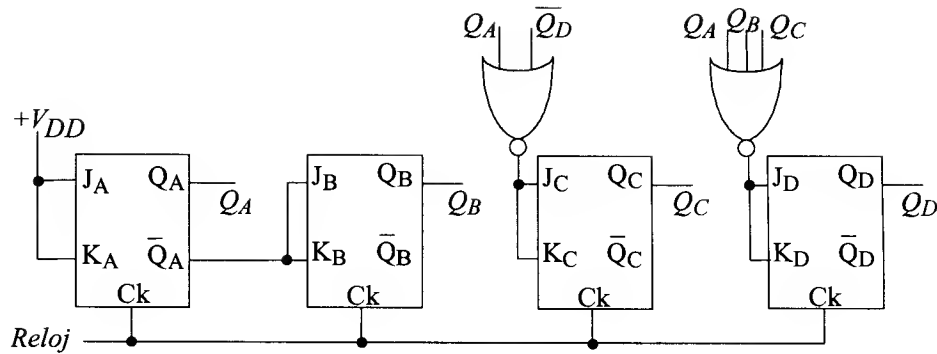
c) ¿Cómo podríamos hacer reversibles las secuencias a) y b)?.

d) En los apartados a) y b) se han propuesto dos secuencias (pares, impares), ¿qué circuitos sería necesario añadir para que bajo el control de una variable externa (S) las recorriera de forma alternativa. Si  $S=0$ , debe recorrer la secuencia de los números pares y cuando  $S=1$ , debe de recorrer la secuencia de los números impares.

**E.9.7.** Analizar los circuitos de las figuras adjuntas. ¿Qué tipo de contadores son?. Dibujar el diagrama de transición de estados y justificar la necesidad de las expresiones lógicas correspondientes a las funciones de excitación (J y K) de cada uno de los cuatro biestables.



(a)



(b)

**E.9.8.** Hemos estudiado los registros de desplazamiento a partir de un ejemplo representativo de las soluciones MSI en tecnología integrada. En este ejercicio queremos abordar su síntesis de forma general, con PLD's y biestables D. Vamos a limitarnos sin pérdida de generalidad a registros de desplazamiento de 3 bits con la estructura de la figura 9.18. Es decir, con tres biestables D conectados en serie, de forma que la salida del primero ataca al segundo y la del segundo al tercero. En paralelo, las tres salidas ( $Q_0$ ,  $Q_1$ ,  $Q_2$ ) junto con la entrada ( $x$ ), participan en una función de lógica combinacional que genera la entrada al primer biestable.

- Suponiendo que  $D_0 = f(x; Q_0, Q_1, Q_2) = x$ , calcular las matrices de transición y la matriz funcional.
- ¿Qué cambios tendríamos que introducir en la matriz funcional si la conexión entre biestables partiera de  $\bar{Q}$ ? Es decir, si

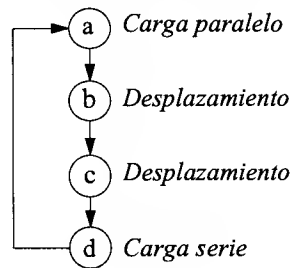
$$D_0 = f(x; Q_0, Q_1, Q_2)$$

$$D_1 = \bar{Q}_0$$

$$D_2 = \bar{Q}_1$$

**E.9.9.** Analizar el registro universal SN74S195 cuyo esquema aparece en la figura 9.34.a. y su cronograma en la figura 9.35. En el desarrollo teórico se ha presentado una descripción cualitativa del funcionamiento, sin obtener las expresiones lógicas de  $R_A$ ,  $R_B$ ,  $R_C$  y  $R_D$  para las situaciones de carga serie, carga paralelo y desplazamiento serie. Obtenerlas aquí, comprobando la tabla de la figura 9.34.b.

**E.9.10.** *Diseñar un circuito que controle el registro de desplazamiento del problema anterior haciéndolo pasar sucesivamente por los siguientes modos de funcionamiento:*



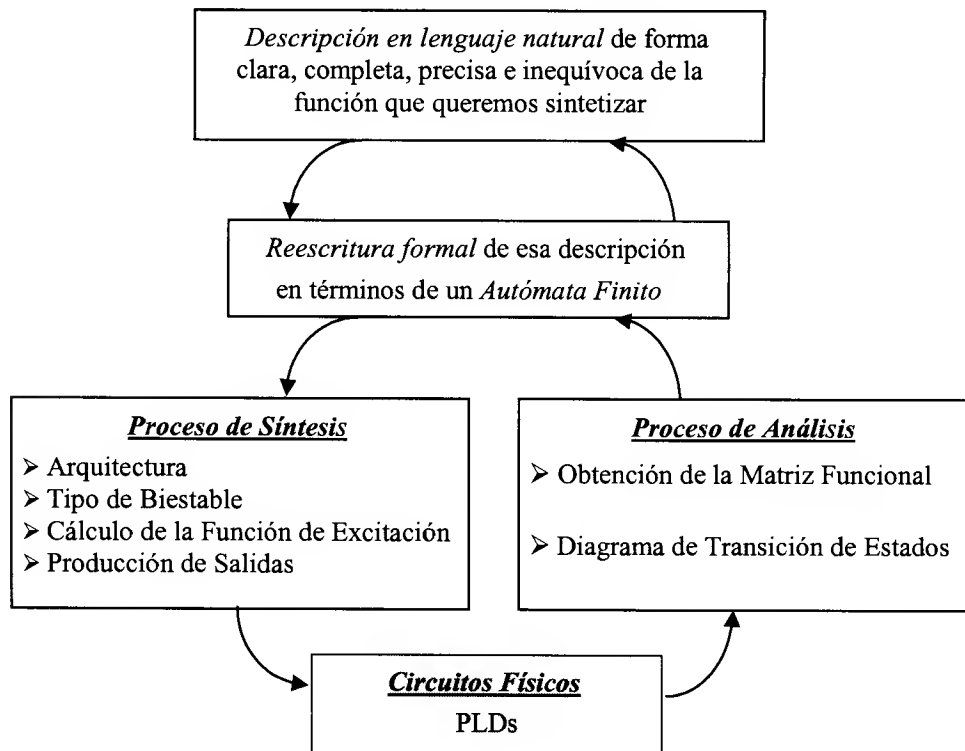
# Preparación de la Evaluación

## Objetivos

- 1: Circuitos secuenciales
- 2: Contadores
- 3: Registros de desplazamiento

## 1: Circuitos secuenciales

- 1.1. El esquema adjunto resume de forma cualitativa el procedimiento general de síntesis y análisis de circuitos secuenciales. ¿Podría identificar los distintos pasos apoyándose en el ejercicio E.9.1 del texto de problemas sobre el control de un ascensor?





Este ejercicio es extenso y lo hemos puesto sobre un problema cuya solución conoce el alumno porque no importa la solución en sí, sino el procedimiento usado para obtenerla. Es decir, el alumno tiene que pensar cuantas historias diferentes de estímulos aparecen, cómo asignarle estados, cómo construir las matrices de transición y/o la matriz funcional, etc... y todo de forma *precisa e inequívoca*.

- 1.2.** *Completar ahora el ejercicio anterior en la tarea de análisis. Partir de la solución que se ofrece en el ejercicio E.9.1 del texto de problemas y obtener la descripción formal.*

*Reflexionar sobre las dificultades de la última fase de este camino inverso. Es decir, sobre la dificultad de pasar desde el autómata a la descripción en lenguaje natural. En realidad no le costará mucho darse cuenta de que tal cosa no es posible, porque un mismo autómata puede representar el substrato formal de muchos problemas descritos en lenguaje natural.*

- 1.3.** *El ejercicio E.9.4 pertenece a una familia general de problemas en los que se necesita un circuito secuencial que detecte una determinada secuencia temporal en los valores de una señal digital. Estos problemas se pueden resolver, al menos, de dos formas:*

- a)** *Usando registros de desplazamiento de carga serie y salida paralelo para tener accesibles todos los bits de la subsecuencia que queremos detectar. La configuración pedida se detecta usando lógica combinacional a partir de los estados,  $\{Q_i\}$ , de los biestables D que constituyen el registro.*
- b)** *Por el procedimiento general de síntesis de autómatas finitos, identificando primero el número de estados necesarios y buscando después el diagrama de transición de estados y la matriz funcional (síntesis con PLDs) o pasando directamente al cálculo de las funciones de excitación de los biestables J-K necesarios para conseguir esos estados y sus transiciones (método convencional).*



*Revisar ese ejercicio observando cómo se hace la síntesis del detector de la secuencia 111 (sin solapamiento) y proponer ahora la solución para un circuito que detecte las siguientes secuencias:*

- I.** *000 (sin solapamiento)*
- II.** *111 (con solapamiento, es decir, siempre que hayan tres unos juntos)*

¿Podría esbozar una ley general para las 8 secuencias de 3 bits (000, 001, 010,...,111), con y sin solapamiento?.

## 2: Contadores

- 2.1. En el texto hemos descrito la clasificación usual de los distintos tipos de contadores básicos: Asíncronos o síncronos, reversibles o no reversibles, binarios o divisores por  $k \neq 2n$ , con arrastre serie o paralelo, etc... Reconstruya el árbol de clasificación y describa con precisión los distintos tipos finales (asíncronos - no reversibles - binarios, ...)
- 2.2. Diseñar un contador asíncrono binario de 3 bits usando biestables J-K disparados a bajadas ( $\downarrow$ ). Dibujar el cronograma correspondiente.
- 2.3. ¿Cómo se modificaría el cronograma del problema anterior si los biestables se dispararan en la subida de los pulsos ( $\uparrow$ ) en vez de en las bajadas ( $\downarrow$ )?. ¿Qué nueva modificación se obtendría si para los biestables del segundo y del tercer bit tomamos como entrada de reloj las señales de  $\overline{Q_0}$  y  $\overline{Q_1}$  en vez de tomarlas de  $Q_0$  y  $Q_1$ ?
- 2.4. ¿Podría ahora resumir los cuatro ciclos de contar entre 0 y 7 de acuerdo con la siguiente tabla?

	tomo de $Q$ la entrada del reloj	tomo de $\overline{Q}$ la entrada del reloj
Disparo a 	?	?
Disparo a 	?	?

- 2.5. ¿Cómo podría complementar alguna de las cuatro opciones del problema anterior para que el contador nos informara sobre el flujo de coches que han entrado y/o salido de un aparcamiento (hasta 7) y de los que hay dentro en un determinado instante (hasta 7)?.
- 2.6. Modificar el diseño del contador del ejercicio 2.2. para que pase de binario a recorrer ciclos menores de 6, 5, 4 y 3 estados (0,1,2,...,5,6,0: 0,1,2,...,5,0: 0,1,2,3,4,0 y 0,1,2,3,0). ¿Podría

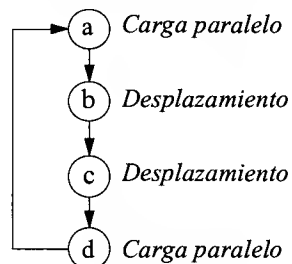
*formular algún tipo de ley general para este tipo de aplicación?. ¿Cree que se facilitaría el diseño si se dotara a los biestables de entradas adicionales de preset y clear?. Demuéstrelo.*

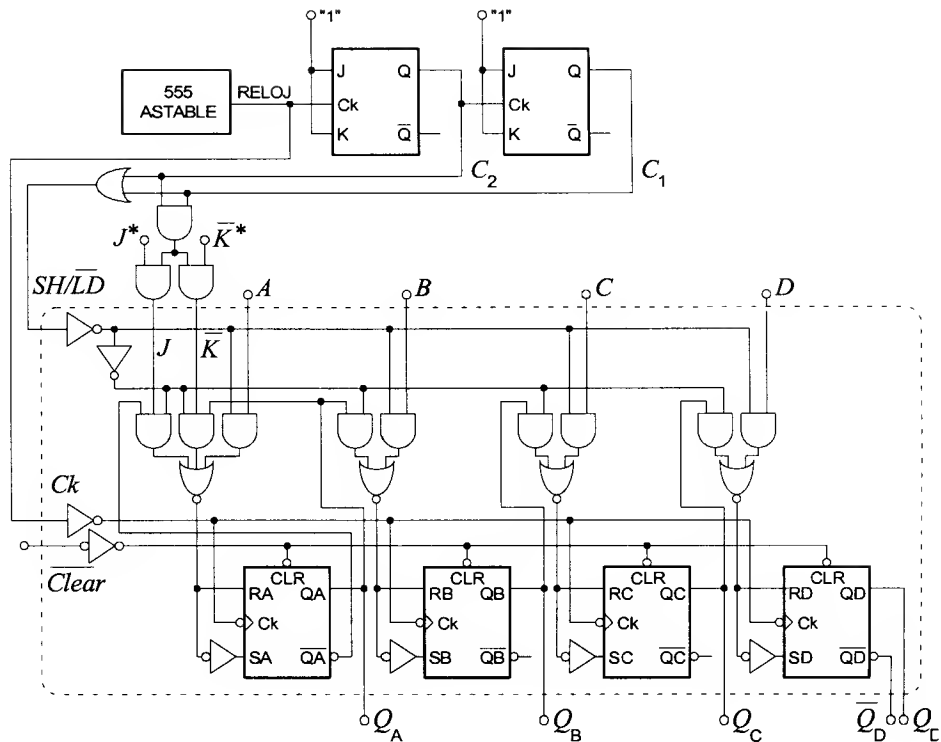
*Le sugerimos que revise el ejercicio E.9.5. del texto de problemas si encuentra dificultades para resolver este ejercicio.*

- 2.7.** *Usar ahora las entradas de Preset y Clear junto con una señal de control de carga (load) para cargar al contador asíncrono de 3 bits con un estado inicial (011), de forma que al final del ciclo vuelva también a ese estado (3,4,5,6,7,3).*
- 2.8.** *Hemos visto en el texto el uso del simulador para comprobar el funcionamiento del contador síncrono 74163 que incluye señales de control de carga, de facilitación de la entrada y de arrastre para conectar con otros contadores análogos. Simular la siguiente secuencia análoga a la que ya hemos visto:*
- a)** *Poner todas las entradas a cero.*
  - b)** *Cargar como estado inicial el número 8 en binario.*
  - c)** *Contar hacia arriba hasta 13.*
  - d)** *Inhibir el contador.*

### **3: Registro de desplazamiento**

- 3.1.** *Revise el ejercicio E.9.9. donde se analiza el funcionamiento del registro de desplazamiento SN74195 y ponga especial atención al control de la carga serie y paralelo. Para ver su funcionamiento es conveniente simularlo siguiendo los pasos análogos a los del problema anterior y así ir viendo cómo actúan cada una de las señales de control.*
- 3.2.** *Analice el circuito de la figura y compruebe que se trata de un registro de desplazamiento que sigue la siguiente secuencia de modos de funcionamiento:*





Revise el ejercicio E.9.10 del texto de problemas, si tiene dificultades.

**3.3.** Diseñar de forma incremental y usando biestables J-K con Preset y Clear asíncronos, un registro de desplazamiento de 3 bits con las siguientes funcionalidades:

- Carga serie y desplazamiento.
- Carga serie, desplazamiento y realimentación controlada por una variable externa ("real"), tal que  $real = 1$  hace recircular la información. Y  $real = 0$ , abre el paso a la nueva carga.
- Que pueda borrarse y cargarse en paralelo y de forma asíncrona.

Usar como fuente de inspiración los dos ejercicios anteriores para resolver este.

**3.4.** Una de las posibles aplicaciones de los contadores y los registros es detectar secuencias binarias. Revise el ejercicio E.9.4. y resuélvalo usando registros de desplazamiento. ¿Podría formular algún tipo de arquitectura general para resolver este tipo de problemas?.

**3.5.** *Otra de las aplicaciones de las estructuras tipo contador o registro es la síntesis de secuencias preespecificadas de estados, correspondan o no a la secuencia de “contar”. Revise el ejercicio E.9.6. para ver la síntesis de las secuencias de números pares e impares y proponga un circuito que sintetice la secuencia: 0 1 2 3 7 6 5 0.*



En este tema hemos partido de módulos pequeños (puertas y biestables) y hemos estudiado las arquitecturas de conexión de esos módulos para obtener funciones de complejidad y grado de integración intermedios. A partir de ahora se amplía nuestro repertorio de componentes reutilizables para el diseño secuencial. No sólo podremos seguir usando las *puertas*, los *MUX*, las *ALUs* y los *biestables*, sino que los *contadores* y *registros* también están a nuestro alcance como “*elementos de circuitos*”, es decir como *bloques funcionales básicos* para la síntesis de otras funciones. Por ejemplo, para direccionar memorias, para controlar procesadores, para facilitar las transferencias entre registros, para controlar la entrada y salida de operandos a una ALU, etc...

Esta es la idea del diseño jerárquico recursivo que subyace en toda la Electrónica Digital. De hecho, el propio microprocesador termina siendo un elemento de circuito.

## REFERENCIAS BIBLIOGRÁFICAS

1. Moreno-Díaz, R.: “*Realizability of a Neural network Capable of all Possible Modes of Oscillation*” en E. Caianiello (ed), “*Neural Network*”. Springer-Verlag. Berlin, 1968.
2. Texas Instruments. “*The TTL Data Book*”. 1987.
3. [www.fairchildsemi.com](http://www.fairchildsemi.com)
4. [www.onsemi.com](http://www.onsemi.com)



# TEMA 10

---

## ***Temporizadores y relojes***

---

### **+ CONTEXTO**

*Para completar las necesidades del diseño en lógica secuencial nos hace falta estudiar los temporizadores y los relojes. A lo largo de todos los temas anteriores, y en particular en el diseño secuencial, siempre hemos supuesto que había un circuito que producía pulsos individuales de distinta amplitud y duración y trenes de impulsos para controlar a bajadas ( $\downarrow$ ) o a subidas ( $\uparrow$ ) las transiciones de estado en los biestables con los que diseñábamos los contadores y los registros .*

*En este tema vamos a abordar el estudio de este tipo de circuitos haciendo distinción, como siempre, entre el comportamiento externo observable y la estructura interna que, conectando puertas, comparadores, biestables y contadores, posibilitan ese comportamiento.*

## ♦♦ CONOCIMIENTO PREVIO NECESARIO

*Todo lo necesario de la Electrónica Digital ha sido estudiado en capítulos anteriores (puertas, biestables y contadores). Hay una pequeña parte de Electrónica Analógica necesaria para calcular la duración de los pulsos (en los circuitos monoestables) y la frecuencia de un tren de impulsos (en los circuitos astables) a partir del proceso de carga y descarga de un condensador que no se ha visto previamente, pero es tan sencilla que no creemos necesario ningún conocimiento previo específico. Lo mismo ocurre en el caso de los osciladores que usan un cristal de cuarzo para fijar la frecuencia de oscilación. El conocimiento que se ofrece aquí sobre este aspecto del tema es muy superficial y cualitativo.*

*Además, como hemos comentado en otras ocasiones, esta asignatura no es una isla en el plan de estudios. El alumno suele estudiar a la vez asignaturas sobre las Bases Físicas de la Informática, donde encontrará los conocimientos de análisis de circuitos necesarios y suficientes para este capítulo.*

## ♦♦♦ OBJETIVOS DEL TEMA

*Los objetivos se centran en estudiar los distintos tipos de circuitos encargados de manejar el tiempo en los sistemas digitales. Al terminar el capítulo, el alumno deberá poder manejar con soltura los siguientes aspectos:*

- Objetivo 1:** *Comprensión de la estructura interna y del comportamiento de los circuitos monoestables (que poseen un estado estable y otro metaestable de duración controlable) y de los circuitos astables (que poseen dos estados metaestables entre los que oscila).*
- Objetivo 2:** *Comprender la estructura interna y la posibilidad de programación externa de los circuitos tipo 555 y de otros temporizadores programables*
- Objetivo 3:** *Adquirir el conocimiento general sobre la estructura interna y el comportamiento de los relojes monofásicos y polifásicos.*

## ++++ GUÍA DE ESTUDIO

*En este tema es particularmente adecuado el uso del simulador para comprobar experimentalmente la influencia de los componentes pasivos ( $R_A$  y  $R_B$  en el 555, por ejemplo) en la forma de onda de salida.*

*El contenido del texto de problemas relacionado con este capítulo es particularmente extenso y fue redactado, tras algún tiempo de docencia, teniendo en cuenta las dificultades que nos comentaban los alumnos. Por consiguiente, aconsejamos su estudio con una relación muy estrecha con el texto de teoría porque de hecho, se complementan.*





---

## ***Contenido del Tema***

---

### **10.1. CIRCUITOS DE TIEMPO**

Para completar las exigencias computacionales de la electrónica digital nos hace falta introducir los circuitos de tiempo. A lo largo de todos los temas anteriores siempre hemos supuesto que existía un reloj que producía un tren de impulsos responsable del comportamiento de todos los sistemas síncronos. Ese circuito es básicamente un oscilador no lineal en el que la frecuencia de oscilación la marca un cristal o un circuito multivibrador, llamado también *astable* porque es un circuito binario (de dos estados) en el que ninguno de los estados es estable durante mucho tiempo, de forma que el circuito está cambiando de estado constantemente.

En este tema vamos a hablar de estos circuitos responsables de marcar el tiempo en los sistemas digitales. La *figura 10.1* ilustra el tipo de formas de onda que necesitamos generar y el nombre del circuito que la produce. Hablaremos primero de forma general y después iremos estudiando los circuitos correspondientes en grado de complejidad creciente.

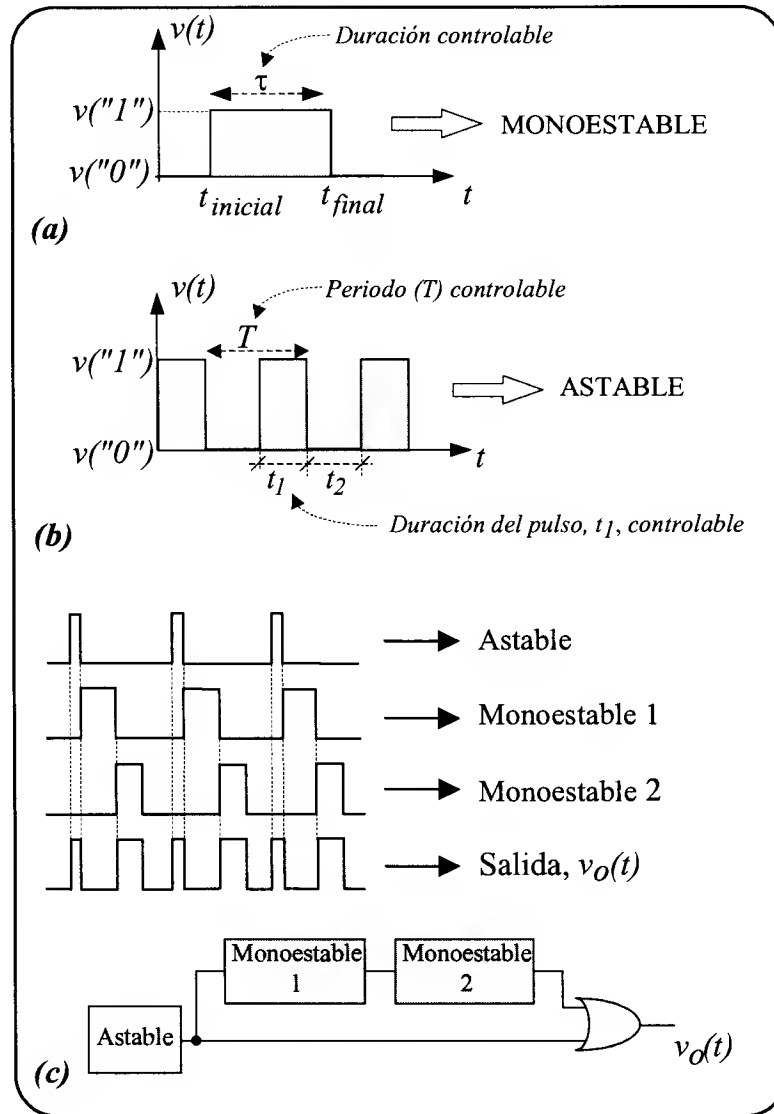
#### **1. *Temporizador elemental.***

Marca un intervalo de tiempo de duración predeterminada y con instantes de inicio y fin bien definidos. El circuito que realiza esta función se llama *monoestable*, porque sólo posee un estado estable (el de baja), siendo la duración del estado de alta la que define la duración del pulso (*figura 10.1.a*). Se usa para producir retardos de valor controlable, para conformar pulsos y para definir sucesos temporales entre dos pulsos sucesivos de un reloj.

#### **2. *Oscilador digital elemental.***

Genera una onda cuadrada o un tren de impulsos de frecuencia controlable. Se llama *astable* o *multivibrador* porque al igual que en los biestables ambos estados eran estables y en los monoestables sólo era estable uno de los estados, en los circuitos astables ninguno de los dos estados es estable de

forma que en su operación normal está conmutando constantemente entre los dos estados (figura 10.1.b). La frecuencia de oscilación la marca la red de acoplo capacitivo que enlaza a dos amplificadores realimentados.



**Figura 10.1.** Necesidades de circuitos de tiempo. (a) Monoestable (genera un pulso de duración predeterminada). (b) Astable (oscilación). (c) Formas de onda compuestas sintetizables a partir de astables, monoestables y puertas lógicas.

### 3. *Formas de onda compuestas.*

Es difícil que una aplicación exija sólo un tren de impulsos (astable) o un pulso de duración controlada (monoestable). Lo usual en los sistemas digitales es que sean necesarias formas de onda más complejas, tal como hemos visto en los cronogramas de los circuitos estudiados en los temas anteriores.

En los sistemas síncronos todas las formas de onda se construyen a partir de un oscilador básico y entre pulsos sucesivos se introducen retardos y se generan otros pulsos más estrechos, por ejemplo, para asegurarnos que la lectura de un dato se realiza cuando ciertos niveles están estables. En el ejemplo propuesto en la *figura 10.1.c* para ilustrar este tipo de señal se dan "pistas" para su posible síntesis usando un astable y dos monoestables en la conexión propuesta. Obsérvese que la señal  $v_O(t)$  es el dato, lo que necesitamos y el circuito que combina el astable y los dos monoestables a través de una puerta OR, es la solución.

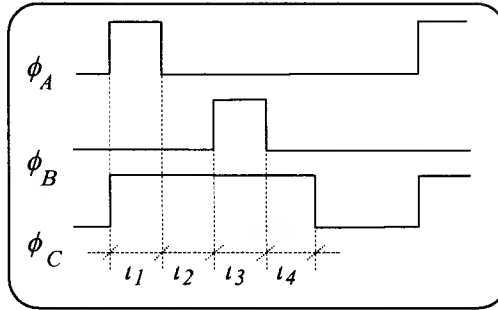
En general, para la síntesis de estas señales compuestas se usan composiciones de monoestables y astables obtenidas a partir de temporizadores programables. El primer circuito de este tipo fue el 555 introducido por Signetics en 1972. Este circuito puede configurarse como astable o como monoestable y posee facilidades adicionales de reset, control de modo y programación de ambas formas de onda. Una segunda generación de temporizadores programables de la que es representativa la solución ICL-8240, se caracteriza por la inclusión de un contador programable y más facilidades de control.

### 4. *Relojes monofásicos y polifásicos.*

La función de un multivibrador no posee la estabilidad suficiente para hacer depender de ella todo el control de un circuito de la complejidad de un microprocesador, por ejemplo. Por eso existen circuitos de tiempo a los que llamamos relojes en los que la frecuencia de oscilación la fija un cristal. El resto de la electrónica se encarga de hacer compatible los niveles lógicos con la familia en cuestión (ECL, TTL o CMOS), introducir señales de facilitación/inhibición y, en algunos casos, añadir un contador como divisor de frecuencias. Esta descripción corresponde a la de un *reloj monofásico*.

Existen otras aplicaciones, en general relacionadas con la *lógica dinámica* de la que hablaremos en los temas siguientes, en las que hacen falta un conjunto de señales de reloj superpuestas de forma que todos los pulsos que aparecen dentro de un periodo mantengan una relación específica entre ellos. Este es el concepto de *reloj polifásico* que se ilustra en la *figura 10.2*. Con su estudio terminaremos este tema cuyo propósito es mostrar algunos de los circuitos básicos necesarios para la síntesis de

cualquier secuencia de temporización.



**Figura 10.2.** Formas de onda típicas de los relojes polifásicos. Las tres señales,  $\phi_A$ ,  $\phi_B$  y  $\phi_C$ , mantienen una relación fija que se repite en cada periodo.

## 10.2. MONOESTABLES

Un monoestable es un circuito binario con *un estado estable y otro metaestable*. La duración de este último se controla a través de la constante de tiempo de una red R-C que establece la realimentación entre dos inversores. Recordemos que cuando este acoplo entre inversores era sólo resistivo obteníamos un circuito biestable (tema 8). Bajo condiciones estacionarias, un circuito monoestable se encuentra en su estado estable y ahí seguirá indefinidamente si no se le aplica ninguna señal externa. Al llegar una señal de disparo el circuito pasa al estado metaestable y allí permanece hasta que el proceso de carga del condensador lo lleva de nuevo al estado estable. El tiempo que permanece en el estado metaestable ( $\tau = R \cdot C \cdot \ln 2$ ) es controlable a través de  $R$  y  $C$  y marca la duración del pulso.

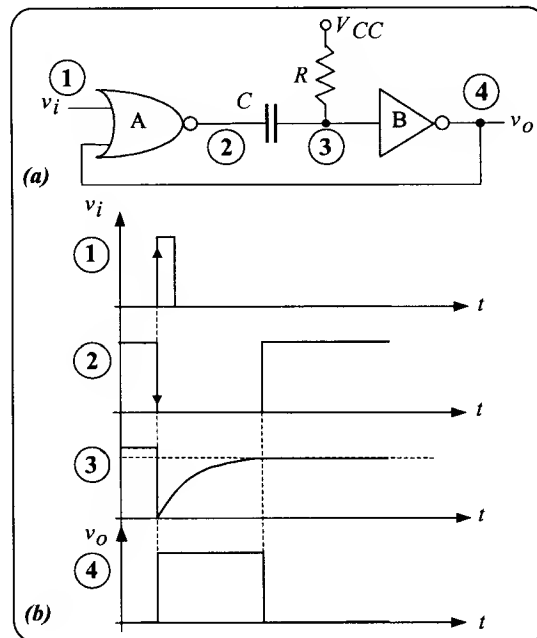
El circuito de la *figura 10.3.a* muestra el esquema de dos inversores acoplados a través de una red R-C y las formas de onda que se generan en distintos puntos importantes del circuito: (1) *Pulso de disparo*, (2) *salida del primer inversor*, (3) *entrada al segundo inversor* y (4) *señal de salida*. La duración del pulso depende de la característica de transferencia del inversor B (CMOS) que fija el punto de disparo en el proceso de carga hacia la tensión de alimentación,  $V_{CC}$  siendo:

$$\tau = -R \cdot C \cdot \ln \frac{V_{CC} - V_T}{V_{CC}}$$

[10.1]

Vamos a describir el funcionamiento del circuito siguiendo el camino de la señal de disparo de

acuerdo con el cronograma de la *figura 10.3.b*. Inicialmente el condensador está descargado ya que tanto la entrada de disparo (1) como la señal de realimentación están en baja y por consiguiente la salida de la puerta NOR está en alta. Tras la aparición del pulso de disparo la tensión en (2) cae abruptamente y el condensador C transmite el impulso a la entrada del inversor B a la vez que empieza a cargarse hacia  $V_{CC}$ . En consecuencia el inversor B pasa a alta y ahí se mantiene hasta que la tensión de carga del condensador atraviese el valor umbral,  $V_T$ , provocando de nuevo el cambio en el inversor B que, junto con la desaparición del impulso de disparo, vuelven la salida de la puerta NOR a su estado inicial de alta.

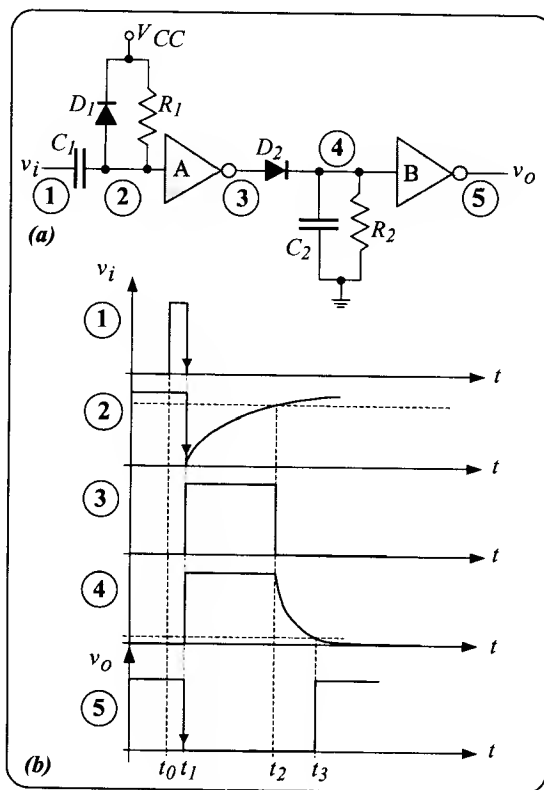


**Figura 10.3.** Diseño de monoestables a partir de inversores CMOS. *(a)* Circuito básico. *(b)* Cronograma, ilustrando la evolución temporal de los potenciales en los distintos puntos de interés para comprender el funcionamiento del circuito.

Existen alternativas más estables a expensas de usar dos constantes de tiempo y, en algunos casos, un diodo adicional. La *figura 10.4.a*, muestra un circuito con dos inversores que se dispara a bajadas.

En el estado estable la entrada al inversor A está en alta y la salida en baja. Por consiguiente, la salida del inversor B está en alta. Al aplicar una transición negativa o un impulso en espiga, tal como

muestra la curva 1 del cronograma de la *figura 10.4.b* el condensador  $C_1$  se carga con polaridad negativa a tierra y la salida del inversor A pasa a alta.  $C_2$  se carga entonces hacia  $V_{CC}$  a través del diodo  $D_1$  y el inversor A, y la salida de B pasa a baja. A medida que  $C_1$  se descarga negativamente se carga a través de  $R_1$  a  $V_{CC}$  (forma de onda 2). La salida de A permanece en alta hasta que la forma de onda de la tensión generada por la carga de  $C_1$  pasa a través del punto umbral de la característica de transferencia del inversor A. En este momento su salida pasa a baja.



**Figura 10.4.** Otra solución al diseño de monoestables con inversores. (a) Circuito. (b) Cronograma (ver descripción en el texto).

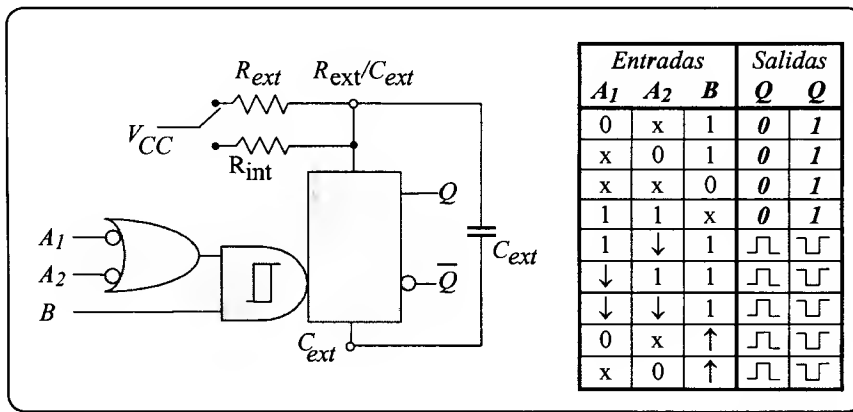
El diodo  $D_2$  evita de forma transitoria la descarga de  $C_2$  que se habría cargado cuando A estaba en alta (forma de onda 3 de 10.4.b). El condensador  $C_2$  empieza entonces a descargarse hacia tierra a través de  $R_2$  (curva 4 de la *figura 10.4.b*). La salida de B permanece en baja hasta que la forma de onda generada por la descarga de  $C_2$  pasa por el punto umbral de la característica de transferencia de

B. En este punto la salida de B vuelve al estado de alta característico del estado estable inicial (curva 5 de la figura 10.4.b).

La ventaja de usar dos puertas conectadas como inversores y pertenecientes a un mismo circuito es la similaridad de sus características de transferencia. Estos ejemplos de diseño de monoestables usando inversores o puertas NOR no tienen más interés que el pedagógico. Al permitirnos "meternos dentro" del monoestable facilitamos la comprensión de su principio de funcionamiento. Sin embargo, hay soluciones integradas en TTL (SN74121 y 123) y CMOS (SCL4047B) que realizan la función del monoestable con el sólo uso de una resistencia y un condensador externos, incluyendo además otras facilidades adicionales tales como capacidad para inhibir el pulso y disparo a través de un circuito de Schmitt.

Un disparador de Schmitt es un circuito binario que presenta histéresis en su característica de transferencia. Cuando la tensión de entrada va subiendo el umbral de disparo es mayor que en la transición inversa, cuando va bajando. Esta histéresis se aprovecha en el disparo de monoestables y en otros muchos circuitos digitales para eliminar "rebotes".

La *figura 10.5* muestra el diagrama interno, los terminales a los que se conectan la  $R$  y la  $C$  externas y la tabla de disparo del circuito SN74121. Obsérvese que una vez disparado la salida es independiente de la entrada. Si no se conecta ninguna capacidad externa, la duración del pulso es de 30 a 35ns., pero puede variarse entre 40ns y 28s mediante una elección adecuada de la resistencia y la capacidad externas.



**Figura 10.5.** Circuito monoestable en TTL, SN74121 y tabla de verdad. Obsérvese que externamente sólo añadimos Rext y Cext para fijar la duración del pulso.

Para algunas aplicaciones pueden ser más adecuadas las soluciones SN74122 ó 123 (dobles). Estos circuitos son redispares y se disparan a niveles de forma que antes que acabe el estado metaestable pueden ser disparados de nuevo prolongando así la duración del estado y produciendo pulsos de salida muy anchos. También tienen la posibilidad de acortar la duración del estado metaestable a través de una entrada de puesta a cero. Así, hay tres modos de controlar la duración del pulso:

- Selección de la red R-C externa.
- Redisparo, para alargar el pulso.
- Puesta a cero, para acortarlo.

Las figuras 10.6 y 10.7 muestran el diagrama funcional, la tabla de control y un cronograma que ilustra los modos de control de la duración del pulso por redisparo y por puesta a cero. La forma básica de controlar la duración del pulso es mediante la red R-C externa. Para valores de  $C$  superiores a  $10^3 pF$ , el valor del ancho del pulso  $t_w$ , se aproxima por una expresión de la forma  $t_w = 0,28 \cdot R \cdot C \cdot (1 + 0,7/R)$ .

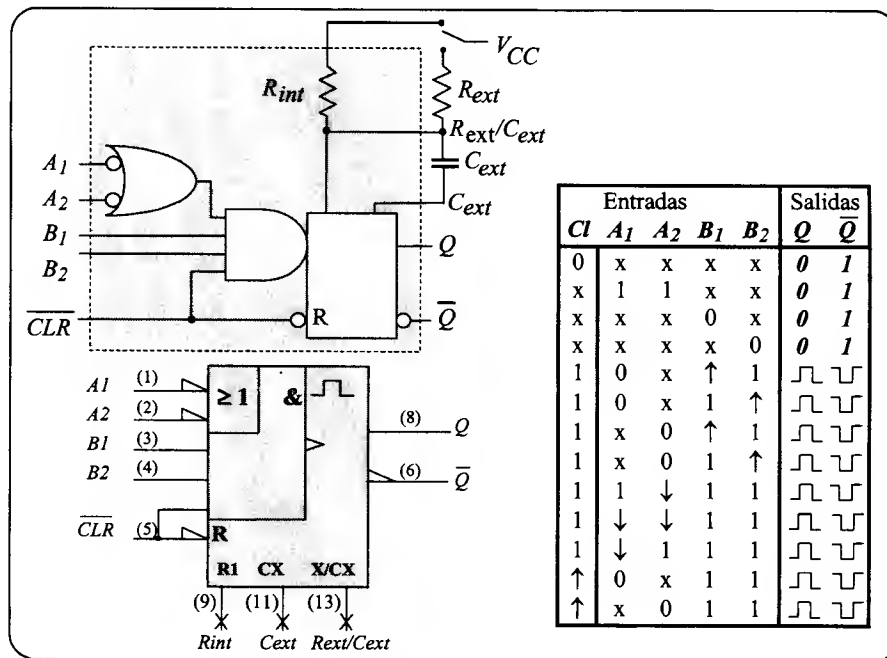
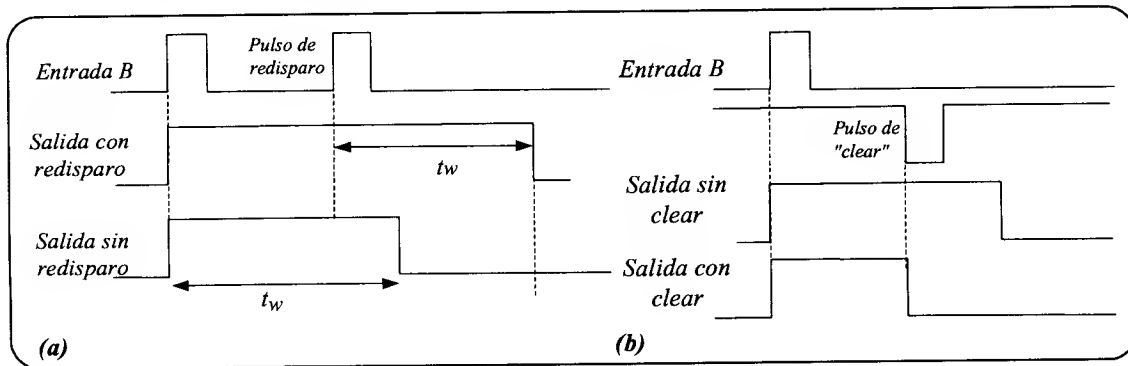


Figura 10.6. Circuito interno del monoestable redispable, SN74122, símbolo lógico y tabla de verdad.



Comparando este circuito con el 121 de la *figura 10.5* se observa una nueva entrada *B2*, la posibilidad de puesta a cero (*clear*) y la configuración que permite prolongar el pulso.



**Figura 10.7.** Cronogramas de los circuitos monoestables redispalables. **(a)** Cronograma de disparo, mostrando la diferente duración del pulso con redisparo y sin redisparo. **(b)** Cronograma de borrado ("clear").

### 10.3. ASTABLES

Tal como comentábamos en la introducción al tema, los circuitos astables son circuitos binarios (de dos estados) en los que ninguno de los estados es estable, de forma que actúan como osciladores no lineales o multivibradores conmutando constantemente entre ambos estados a una frecuencia determinada por la constante de tiempo de la red de acoplo.

El principio de diseño se basa de nuevo en el proceso de carga y descarga de un condensador externo,  $C$ , a través de una resistencia,  $R$ , hacia los valores extremos de la tensión de alimentación. La *figura 10.8* muestra el circuito y las formas de onda correspondientes al diseño de un astable usando un amplificador operacional

Supongamos que inicialmente el amplificador está saturado a  $+V_{CC}$ . Aparecerá entonces a través del partidor ( $R_1$ ,  $R_2$ ) un nivel de tensión en la entrada no inversora de valor:

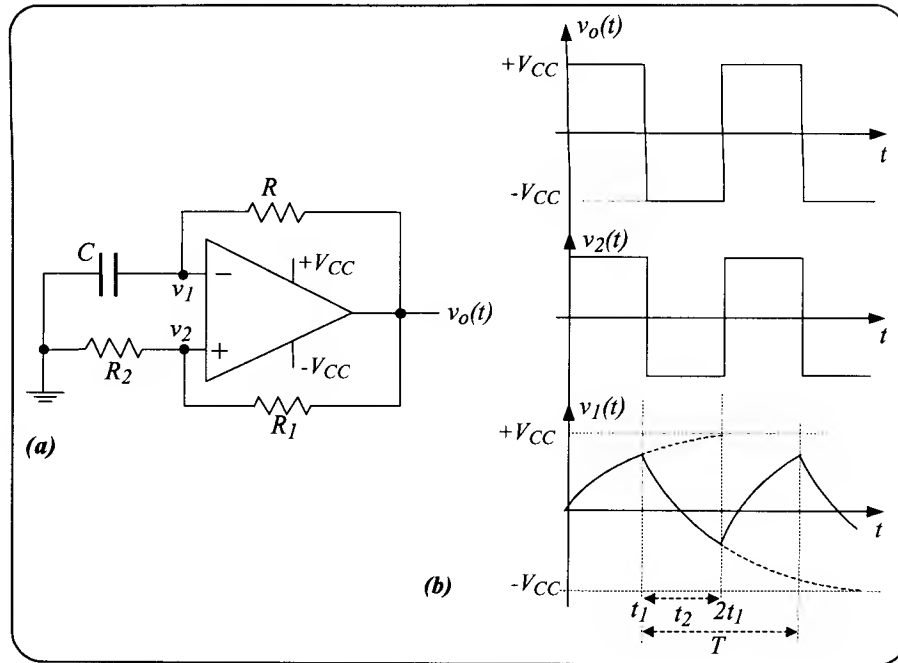
$$v_2 = V_{CC} \frac{R_2}{R_1 + R_2} \quad [10.2]$$

A partir de este instante el condensador se va cargando a través de  $R$  hacia  $V_{CC}$  hasta que el potencial en  $v_1$  supera al valor fijado por el partidor ( $R_1$ ,  $R_2$ ) en  $v_2$ . Así, la tensión  $v_1$  varía según la

expresión:

$$v_I(t) = V_{final} + (V_{inicial} - V_{final}) \cdot e^{-\frac{t}{\tau}} \quad [10.3]$$

$$\text{siendo: } \tau = R \cdot C, \quad V_{final} = +V_{CC} \quad \text{y} \quad V_{inicial} = -V_{CC} \frac{R_2}{R_1 + R_2} \quad [10.4]$$



**Figura 10.8.** Circuito multivibrador o astable. (a) Síntesis mediante un amplificador operacional. (b) Formas de onda en los principales puntos de interés para comprender el funcionamiento del circuito. Obsérvese como la conmutación está asociada al proceso de carga y descarga del condensador C.

En  $t=t_1$  la tensión en la entrada inversora alcanza el mismo valor que el fijado en la entrada no inversora y la salida del comparador diferencial conmuta. Podemos calcular el tiempo que tarda la tensión  $v_I$  en alcanzar el valor de  $v_2$  igualando esta tensión con la expresión de  $v_I(t)$  para  $t=t_1$ . Así,

$$v_I(t) = v_I(t_1) = +V_{CC} + (-V_{CC} \frac{R_2}{R_1 + R_2} - V_{CC}) \cdot e^{-\frac{t_1}{RC}} = V_{CC} \frac{R_2}{R_1 + R_2} \quad [10.5]$$

Despejando  $t_1$  obtenemos: 
$$t_1 = R \cdot C \cdot \ln \left( 1 + \frac{2R_2}{R_1} \right) \quad [10.6]$$

Una vez que conmuta el comparador diferencial su salida pasa a saturación negativa ( $-V_{CC}$ ) y establece un nuevo valor en  $v_2$ :

$$v_2 = -V_{CC} \frac{R_2}{R_1 + R_2} \quad [10.7]$$

Comienza entonces la descarga del condensador desde el nivel positivo anterior hacia  $-V_{CC}$  hasta que cruza de nuevo el valor de  $v_2$  y se repite el ciclo. En el tramo descendente, durante la descarga del condensador, la expresión de la tensión en la entrada inversora es la misma de la expresión [10.3], pero los valores iniciales y finales son distintos. Así, ahora

$$V_{final} = -V_{CC} \quad y \quad V_{inicial} = +V_{CC} \frac{R_2}{R_1 + R_2} \quad [10.8]$$

La constante de tiempo es la misma,  $\tau = RC$ , puesto que la descarga la hace también a través de la misma resistencia,  $R$ , y el valor en el que se produce la conmutación es el nuevo valor de  $v_2$ . Así, podemos calcular el tiempo que tarda en producirse de nuevo la conmutación,  $t_2$ ,

$$v_1(t) = v_1(t_2) = -V_{CC} + \left[ V_{CC} \frac{R_2}{R_1 + R_2} - (-V_{CC}) \right] \cdot e^{\frac{-t_2}{RC}} = -V_{CC} \frac{R_2}{R_1 + R_2} \quad [10.9]$$

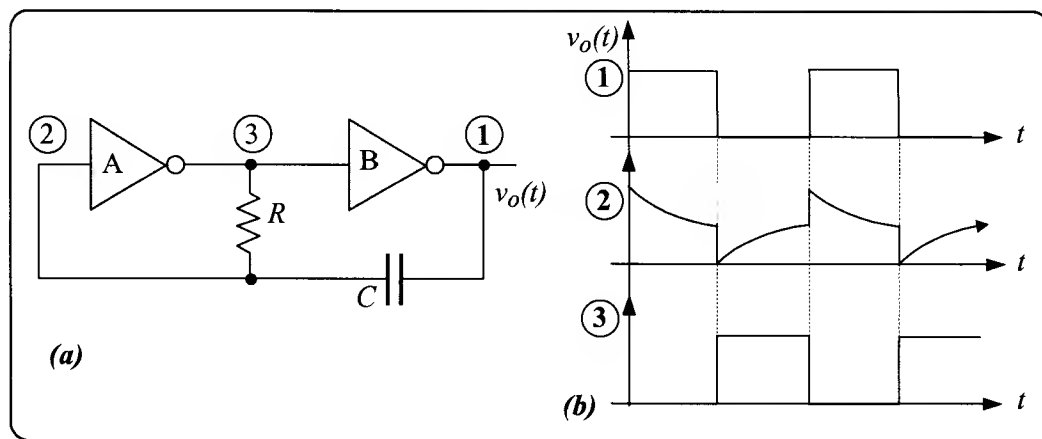
Despejando  $t_2$  obtenemos: 
$$t_2 = R \cdot C \cdot \ln \left( 1 + \frac{2R_2}{R_1} \right) \quad [10.10]$$

Como podemos observar los tiempos  $t_1$  y  $t_2$  tienen el mismo valor lo cual era de esperar ya que el circuito es simétrico . Por tanto el periodo de la señal que genera el astable es:

$$T = 2 t_1 = 2 R \cdot C \cdot \ln \left( 1 + \frac{2R_2}{R_1} \right) \quad [10.11]$$

Al igual que en los otros circuitos binarios, siempre podemos usar inversores integrados en cualquier tecnología para realizar circuitos astables añadiéndole la red R-C externa. El principio de funcionamiento de estos astables es la detección del potencial umbral de las puertas en los procesos de

carga de un condensador que experimenta así un cambio brusco en el potencial de sus extremos. La figura 10.9 muestra una configuración posible.



**Figura 10.9.** Síntesis mediante dos inversores acoplados de un astable. (a) Circuito. (b) Cronograma.

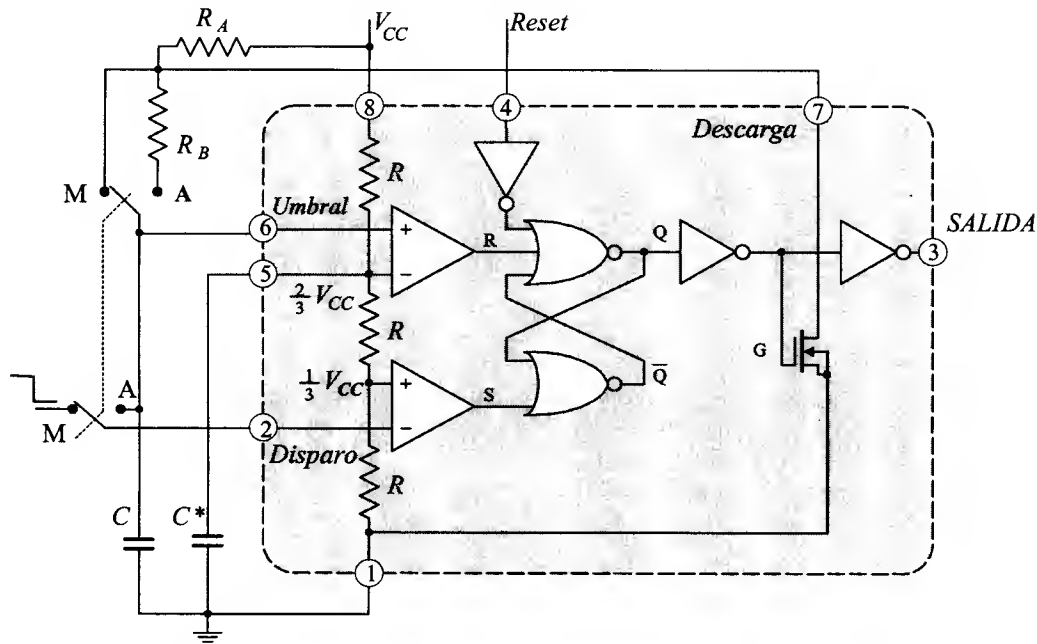
Cuando la señal de salida del inversor B está en alta, su entrada está en baja y el condensador  $C$  está cargado positivamente y la entrada al inversor A es alta. Por consiguiente su salida es baja, de forma que el condensador posee una vía de descarga hacia tierra a través de  $R$ . La descarga de  $C$  (forma de onda 2 de la figura) continúa hasta que la tensión en la entrada de la puerta A pasa por su valor umbral, pasando la salida de A a alta y la de la puerta B a baja. Se inicia así el proceso inverso de carga de  $C$  hasta el siguiente paso por la tensión umbral de la puerta A.

## 10.4. CIRCUITOS DE TIEMPO TIPO 555

Estos circuitos de tiempo fueron introducidos por Signetics en 1972 y se han convertido en la actualidad en un bloque funcional de uso muy general tanto en la Electrónica Analógica como en la Digital. Es capaz de actuar como *monoestable* y como *astable* con duración de pulso y frecuencia de oscilación controlables, realizando también funciones de modulación en anchura de impulsos, detección de pulsos omitidos, etc...

La figura 10.10 muestra el diagrama funcional y las redes externas necesarias para su operación en los modos astable y monoestable. El paso de una configuración a otra se realiza mediante un conmutador de dos posiciones y dos circuitos. Este circuito se realiza en tecnología bipolar y CMOS

(tipo ICM7555) aunque el diagrama funcional es el mismo. La solución CMOS se caracteriza por su bajo consumo.



**Figura 10.10.** Temporizador 555. Circuito interno y esquemas externos de conexión para operación en modo astable (A) y monoestable (M).

Básicamente, el 555 consta de dos comparadores internos que controlan el estado de un biestable el cual a su vez ataca a un transistor de descarga de una etapa de salida. Las tensiones de referencia de los comparadores son siempre proporcionales a la fuente de alimentación, ( $V_{CC}$ ), proporcionando así un sincronismo de salida independiente de la alimentación que puede variar entre 5 y 15 voltios. para el caso bipolar ó de 2 a 18 voltios para el caso CMOS. Veamos a continuación el significado funcional de cada uno de los terminales.

1. *Tierra:* Se conecta generalmente a tierra y su tensión debe ser la más negativa del circuito.
2. *Disparador:* Terminal sensible a niveles de tensión con un valor crítico de  $V_{CC}/3$  que coincide con la señal de referencia interna del comparador 2. Cuando la señal de este terminal baja de  $V_{CC}/3$  el biestable pasa a alta haciendo que la salida en el terminal 3 pase a alta también. A este terminal se pueden aplicar tensiones cuyo valor se encuentre entre el valor de la aplicada en el

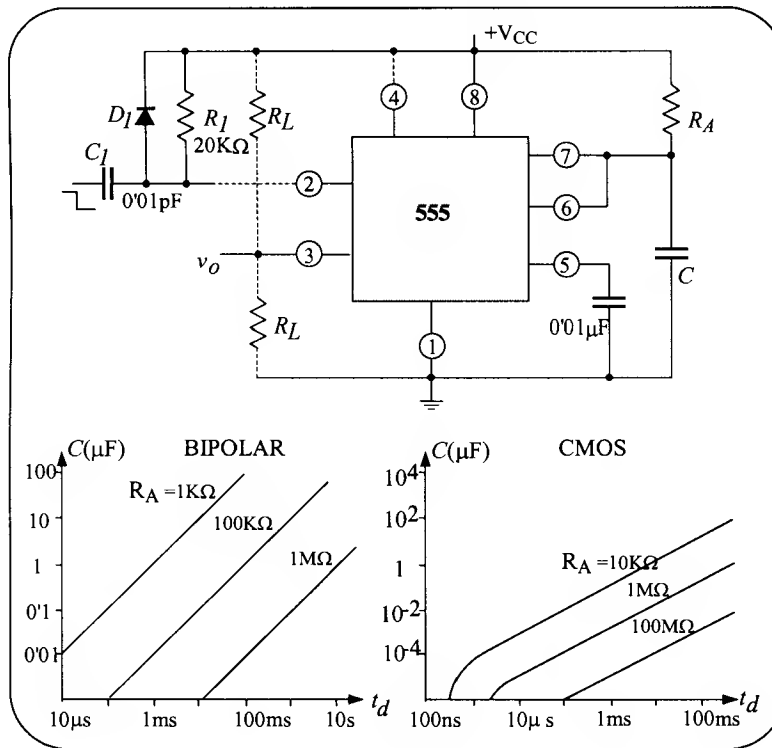
terminal 8 y tierra (terminal 1).

3. *Salida:* El nivel de tensión aquí está generalmente en baja y pasa a alta durante el estado metaestable u oscila entre ambos niveles en el modo astable.
4. *Reset:* Este terminal, tanto en el caso bipolar como en el CMOS posee un nivel de control de 0,6 a 0,7 *voltios*. Cuando la tensión es inferior a este valor se interrumpe el ciclo de sincronismo volviendo el circuito al estado de reposo, pasando a conducción a través del biestable, al transistor que se usa para descargar al condensador, impidiendo que este se pueda cargar de nuevo. Este terminal de puesta a cero se debe conectar a la tensión de alimentación cuando no se use ya que mientras el potencial en este punto no supere un cierto valor el circuito no se puede volver a disparar y la salida permanece constante. Existen otras funciones de control de este terminal muy útiles en aplicaciones específicas.
5. *Control:* Está conectado internamente a la tensión de referencia del comparador 1, es decir a  $2V_{CC}/3$ . Sin embargo se puede conectar a tierra a través de una resistencia externa o a una tensión externa que permite cambiar las tensiones de referencia de los comparadores actuando como entrada de señal de control. Cuando no se usa para este propósito se recomienda unir a tierra a través de un condensador del orden de 0,01  $\mu\text{F}$  para filtrar el ruido de la fuente y evitar pulsos de ruido.
6. *Umbral:* Es la entrada externa al comparador 1 que en la otra entrada tiene aplicada una tensión de referencia de valor  $2V_{CC}/3$ . Por eso cuando la tensión en este terminal supera la referencia el biestable y la salida pasan a baja.
7. *Descarga:* Este terminal es el drenador del transistor de descarga. Se usa normalmente para descargar el condensador de sincronismo.

Vamos a describir ahora los dos modos básicos de operación que se seleccionan cambiando de posición al conmutador de (*M*) a (*A*) y viceversa, tal como se ilustra en el circuito de la *figura 10.10*. En el modo *monoestable*, el terminal 7 se une al terminal 6, el 4 al 8 y entre el terminal 8 y el 1 se introduce el partidor clásico de todos los monoestables con  $R_A$  y  $C$ . Inicialmente, el condensador  $C$  está descargado debido al transistor de descarga. Al aplicar un impulso negativo en el terminal de disparo, cuando el potencial de este terminal (2) cae por debajo de la tensión de referencia del segundo comparador ( $V_{CC}/3$ ), este comparador cambia de estado y arrastra con él al biestable que a su vez pasa al corte al transistor, "desconectando" el terminal 7 y permitiendo la carga del condensador. A su vez

el biestable pasa a alta la salida (terminal 3).

Una vez que el biestable ha sido disparado por una señal de entrada, no se puede disparar de nuevo hasta que no finalice el ciclo. La salida está en alta hasta que en el proceso de carga (con una constante de tiempo  $T = R_A \cdot C$ ) la tensión del condensador alcanza el otro umbral,  $2V_{CC}/3$ . Entonces el primer comparador pone a cero al biestable, conecta el terminal 7 y descarga al condensador a la vez que lleva a baja al terminal de salida. La *figura 10.11* muestra la conexión monoestable y el valor del retardo,  $t_d$ , para un cierto valor de la resistencia de carga para una solución bipolar y otra CMOS. La tabla de la *figura 10.12* resume el comportamiento del 555 operando como monoestable



**Figura 10.11.** Conexión monoestable del temporizador 555 y valor del retardo en las soluciones bipolar y CMOS.

Veamos algunas consideraciones de orden práctico referentes a los valores de  $R_A$ ,  $C$  y  $R_L$ , así como a las características del pulso de disparo. En las soluciones bipolares el valor mínimo de  $R_A$  debe ser tal que la corriente de carga inicial no supere un cierto valor que impide el funcionamiento

adecuado del transistor de descarga. Para usar valores pequeños de  $C$ , es aconsejable elegir  $R_A$  lo más grande posible compatible con el nivel mínimo de corriente necesaria en el terminal 6. Para soluciones bipolares debe elegirse  $R_A$  de forma que la corriente inicial se encuentre entre  $1\mu A$  y  $5mA$  con lo que se pueden conseguir pulsos de duración en un rango de  $5 \cdot 10^{-3}$  a  $1s$ .

555 como MONOESTABLE											
Reset ④	Disparo ②	Umbral ⑥ $v_c(t)$	R	S	Q	G	Estado del Transistor	Descarga ⑦	Umbral ⑥ $v_c(t+\Delta t)$	Salida ③	Estado del Monoestable
0	x	x	x	x	0	1	Conducción	Baja Z		0	Estable
1		$V < 2V_{cc}/3$	0	0	0	1	Conducción	Baja Z	$v_c(t) = 0$	0	Estable
1		$V < 2V_{cc}/3$	0	1	1	0	Corte	Alta Z		1	Metaestable
1		$V < 2V_{cc}/3$	1	0	0	1	Conducción	Baja Z		0	Transición
1		$V < 2V_{cc}/3$	0	0	0	1	Conducción	Baja Z		0	Estable
<b>Cálculo de <math>t_1</math></b>			<div style="display: flex; justify-content: space-between; align-items: flex-start; margin-top: 10px;"> <div> <math display="block">v(t) = v_f + (v_i - v_f) e^{-t/\tau}</math> <math display="block">v_f = V_{cc}, \quad \tau = R_A C</math> <math display="block">v_i = 0, \quad v(t_1) = 2V_{cc}/3</math> <math display="block">2V_{cc}/3 = V_{cc} + (0 - V_{cc}) e^{-t_1/R_A C}</math> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-top: 10px;"> <math display="block">t_1 = R_A C \ln 3</math> </div> </div> </div>								

**Figura 10.12.** Tabla resumen del comportamiento del 555 operando en modo monoestable



En tecnología CMOS, todos los niveles de corriente de disparo, puesta a cero, etc. son extremadamente bajos ( $20pA$ , típico) de forma que podemos usar valores de  $R_A$  en el orden de los  $100M\Omega$  para producir tiempos de retardo de  $10s$  con sólo  $100nF$ , por ejemplo. En cuanto a  $C$ , su valor máximo está determinado por las corrientes de pérdidas y el valor mínimo en bipolar es de  $100nF$  y de  $1nF$  en CMOS.

La carga del 555 bipolar puede ser fuente o sumidero de  $200mA$  con tiempos de salida o bajada de  $100ns$ . Es aconsejable introducir una capacidad de  $100nF$  desde el terminal 3 a tierra y si se prevén componentes inductivos en la carga, atacarla a través de un diodo para eliminar las tensiones negativas. En CMOS la máxima corriente suministrada depende de la alimentación. Para  $18 voltios$ , pueden alcanzarse  $10mA$  y hasta  $50mA$  cuando actúa de sumidero.

El ancho mínimo del pulso necesario para producir el disparo varía entre  $100$  y  $10^3 ns$ , y no puede superar el valor  $R_A \cdot C$ . Esta condición produciría intervalos erráticos y se puede evitar diferenciando el pulso de disparo y usando un diodo de sujeción para evitar que los pulsos positivos superen el valor de la fuente de alimentación.

Veamos ahora la configuración *astable*. Por razones pedagógicas repetimos la conexión en la *figura 10.13*. En esta figura se incluye también las graficas de los pares de valores de  $C$  y  $R_A + 2R_B$  para las distintas frecuencias de oscilación. El circuito se redispara él mismo mediante una realimentación entre los terminales 2 (*disparo*) y 6 (*umbral*), de forma que la tensión en los extremos del condensador oscila entre  $V_{CC}/3$  y  $2V_{CC}/3$ . En este modo de operación separamos el terminal 6 del 7 con  $R_B$ .

La *figura 10.14* muestra, en un cuadro resumen, el comportamiento del 555 en configuración *astable*.

El condensador se carga a  $2V_{CC}/3$  a través de  $R_A$  y  $R_B$  y se descarga hasta  $V_{CC}/3$  a través de  $R_B$ . Así, variando la razón entre  $R_A$  y  $R_B$  se controla el ciclo de oscilación. Si llamamos  $t_1$  al tiempo de carga,  $t_2$  al de descarga y  $T$  al periodo tenemos:

$$t_1 = 0,69(R_A + R_B) \cdot C \quad (\text{salida en alta}) \quad [10.12]$$

$$t_2 = 0,69R_B \cdot C \quad (\text{salida en baja}) \quad [10.13]$$

$$T = t_1 + t_2 = 0,69(R_A + 2R_B) \cdot C \quad [10.14]$$

El máximo *ciclo de uso* es: 
$$\frac{t_1}{t_1 + t_2} = \frac{R_B + R_A}{R_A + 2R_B} \quad [10.15]$$

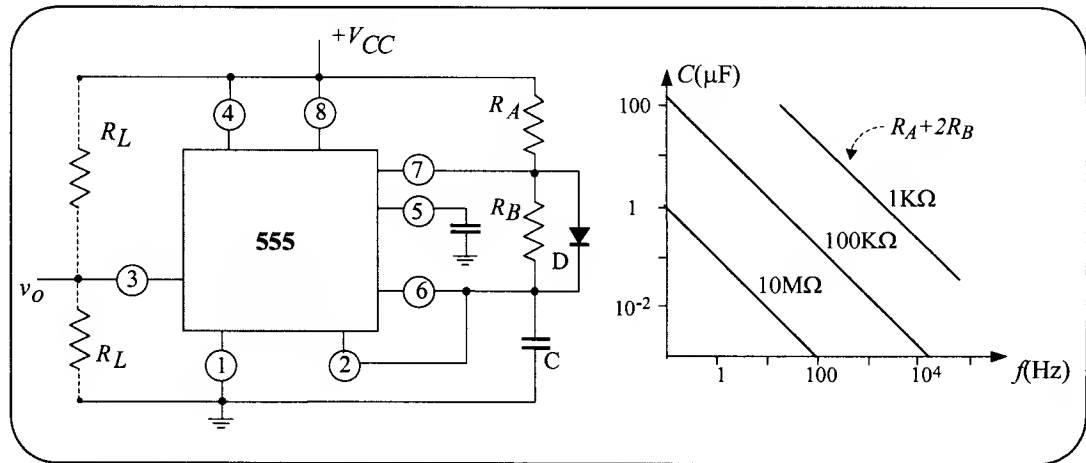
Para hacer máximo este cociente,  $R_A$  debe ser lo más pequeña posible, pero lo suficientemente grande para limitar la corriente en el terminal de *descarga* (7). Para mejorar este ciclo de uso se puede introducir un diodo  $D_1$  en paralelo con  $R_B$ . Así, la corriente de carga pasa a través de  $R_A$  y  $D_1$ , mientras que la descarga es a través de  $R_B$ . En esta configuración los tiempos de alta y baja en la salida son función de resistencias independientes:

$$t_1 \approx 0,69 R_A \cdot C \quad \text{y} \quad t_2 \approx 0,69 R_B \cdot C \quad [10.16]$$

$$T = t_1 + t_2 = 0,69 (R_A + R_B) \cdot C, \quad [10.17]$$

$$\frac{t_1}{t_1 + t_2} = \frac{R_A}{R_A + R_B} \quad [10.18]$$

En la configuración astable se puede usar el terminal de *reset* (4) para llevarlo a tierra cuando se desee parar la oscilación. Tenemos así un control lógico del oscilador con una simple puerta o un conmutador analógico.



**Figura 10.13.** Conexión del 555 en modo astable.

# 555 como A S T A B L E

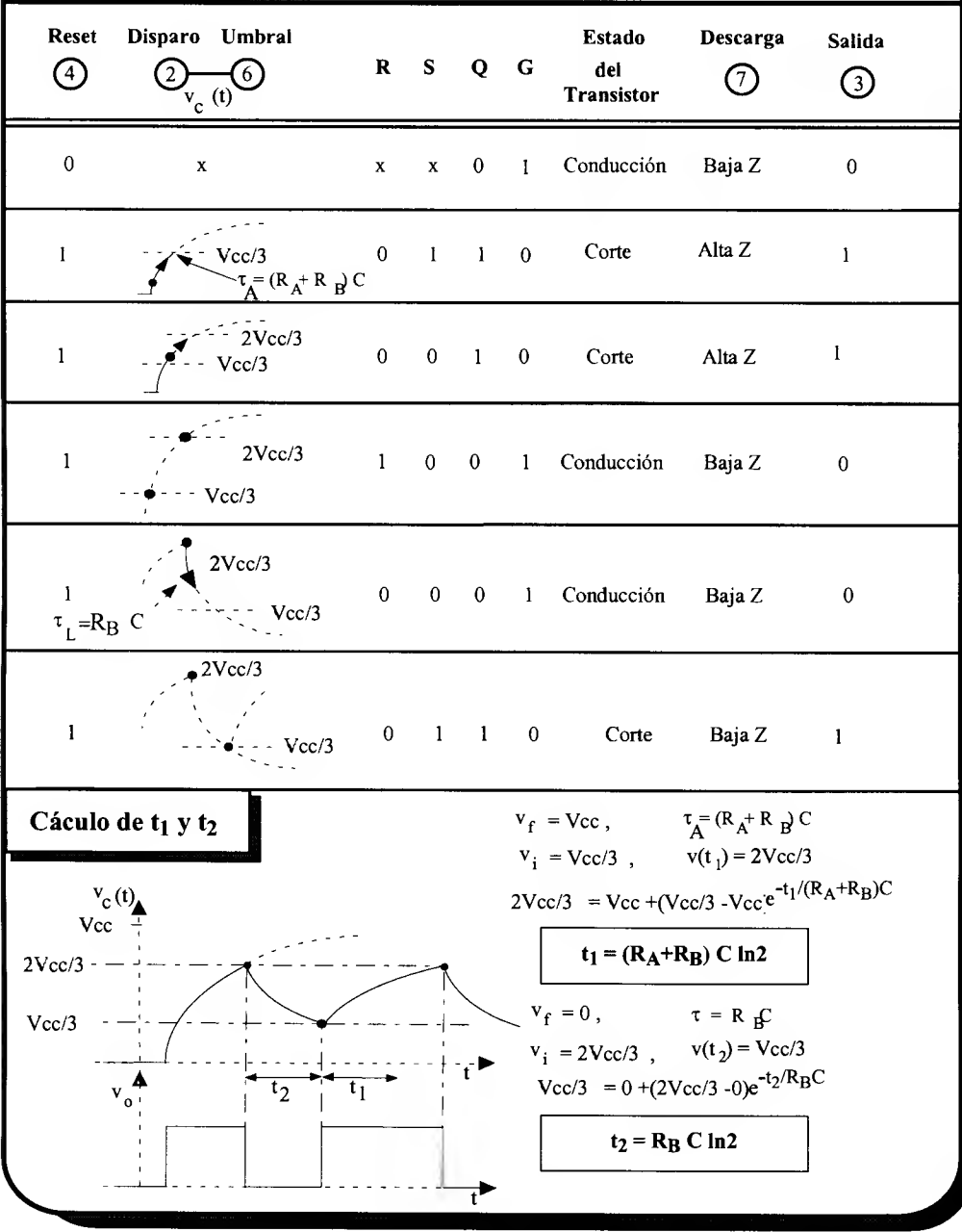
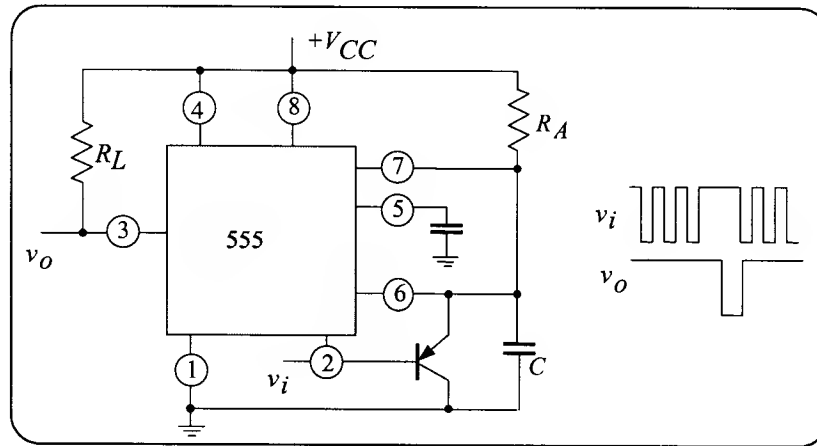


Figura 10.14. Tabla resumen del comportamiento del 555 como astable.

Además de las aplicaciones propias de monoestables y astables, el 555 posee una amplia gama de posibilidades (generación de rampas, modulación de anchura de pulsos, detección de omisión de pulsos, controles de temperatura, ...). A título de ejemplo vamos a comentar el *detector de omisión de pulsos* (figura 10.15) y el generador de secuencias que usamos al comienzo del tema (figura 10.16).

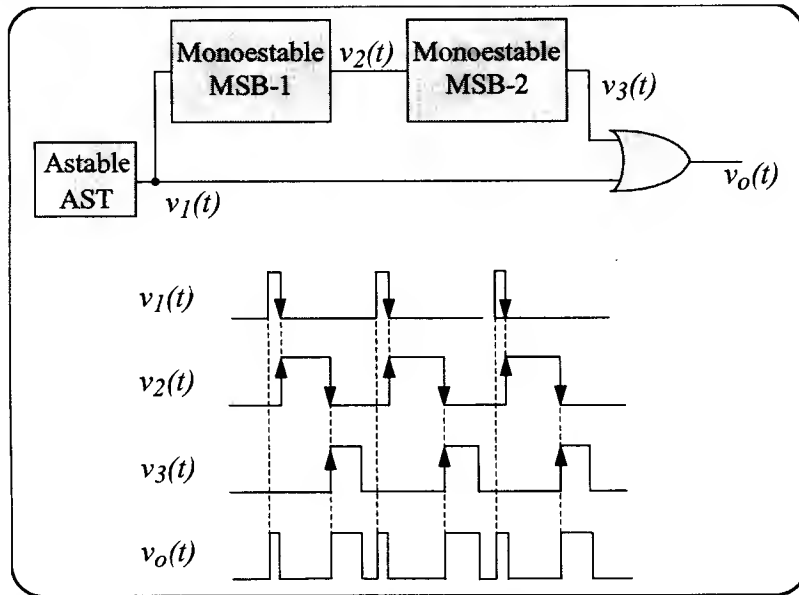


**Figura 10.15.** Ejemplo de aplicación del 555 a la detección de la omisión de un pulso en una secuencia.

Analicemos primero el comportamiento del circuito de la *figura 10.15*. Queremos producir una salida siempre que falte un pulso en la entrada durante el tiempo de retardo. Para conseguir esto, establecemos un tiempo de retardo algo mayor que el intervalo de tiempo existente entre dos pulsos sucesivos de entrada. Así, el ciclo de sincronismo está constantemente puesto a cero por el tren de pulsos de entrada hasta que un cambio en la frecuencia de este tren o la omisión de un pulso permite que, "por fin", se complete el ciclo de sincronismo produciendo así un cambio en el nivel de salida. Es decir, el circuito 555 está funcionando como monoestable redisable ya que los valores de  $R$  y de  $C$  se calculan para que el monoestable esté en alta un tiempo superior al periodo de los pulsos que se quieren detectar. Cuando falta un pulso en la entrada monoestable a este le dá tiempo a volver a su estado estable (salida cero) y no se vuelve a disparar hasta que en la entrada no aparezca de nuevo el flanco negativo del siguiente pulso. Por tanto cada vez que la salida del circuito pasa a cero detecta la falta de pulsos en la entrada.

La *figura 10.16* muestra el ejemplo de diagrama de tiempos que usamos en la introducción del tema y la forma de sintetizarla mediante el uso de circuitos tipo 555 en configuración astable (AST) y monoestable (MSB). Hemos introducido una puerta OR en el ejemplo para señalar una forma de

extender las posibilidades de estos circuitos de tiempo para conseguir secuencias de control.



**Figura 10.16.** Ejemplo de uso del 555 para generación de señales de control arbitrarias.

En el ejemplo, para generar la forma de onda  $v_o(t)$  que es el dato de partida en el diseño, usamos un 555 en configuración *astable* que genera un tren de impulsos. Estos disparan el segundo 555 conectado como *monoestable* (MSB1), quien a su vez dispara al segundo *monoestable*, MSB2. Finalmente, obtenemos la salida deseada,  $v_o(t)$ , sumando las formas de onda del astable,  $v_1(t)$ , y del segundo monoestable,  $v_3(t)$ . Queremos repetir una vez más que el ejemplo sólo pretende ilustrar la posibilidad de realizar señales de control de forma arbitraria con el uso de este tipo de circuito. Alguna "ley" se puede extraer no obstante como generalización del procedimiento de síntesis que hemos usado:

- 1°. Partir de un *oscilador* que genere pulsos "estrechos" y con una frecuencia tal que el intervalo de tiempo entre dos pulsos sucesivos sea suficiente para disparar uno o más monoestables.
- 2°. Usar monoestables para generar *pulsos del ancho deseado* a partir de las referencias temporales que marcan las subidas o bajadas de los pulsos del astable ("reloj").
- 3°. Usar puertas lógicas (AND, OR, MUX y/o DEMUX) para definir formas de onda compuestas, de acuerdo con las necesidades de la aplicación.

- 4°. Todo lo que hemos dicho sólo necesita para su síntesis (aparte del 555), lógica combinacional. Sin embargo, existen muchas funciones de temporización que necesitan "memoria". Para su síntesis se han desarrollado otros tipos de circuitos programables que incluyen, en general, contadores y biestables aislados. De ellos hablaremos a continuación.

## 10.5. TEMPORIZADORES PROGRAMABLES

Para prolongar la duración del retardo y para simplificar y hacer más flexible el diseño con circuitos tipo 555, existe otra familia de circuitos de tiempo (tipo ICL8240, XR2240 ó MM5865), que son programables. Para ello incluyen en el mismo circuito las siguientes funciones (*figura 10.17*):

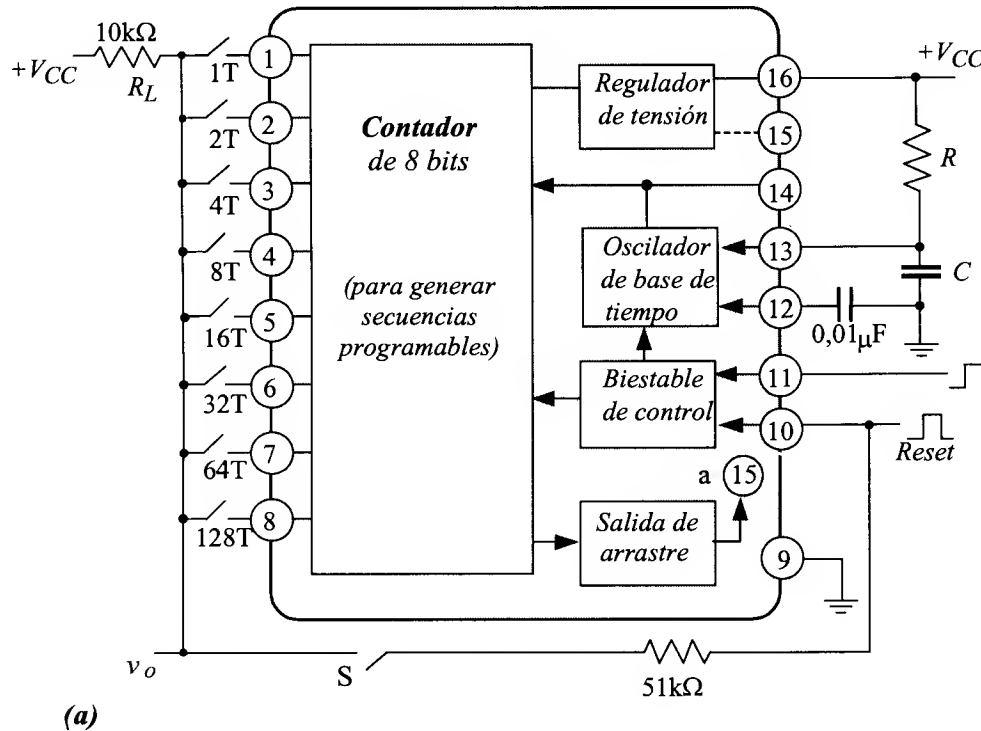
- Un oscilador (base de tiempos).
- Un contador programable.
- Un biestable.
- Circuitos de control.

Estos circuitos pueden operar en modo monoestable o astable. En el modo astable generan la frecuencia base y ocho suboctavas (caso del ICL8240, por ejemplo). En el modo monoestable generan pulsos (tiempo de retardo) que van de microsegundos a horas ó días. La programación de la duración se realiza mediante la selección de una red externa, R-C, y a través de la configuración de vuelta a cero del contador. Para un valor fijo de  $RC$  podemos programar retardos que van desde  $RC$  hasta  $255 \cdot R \cdot C$ .

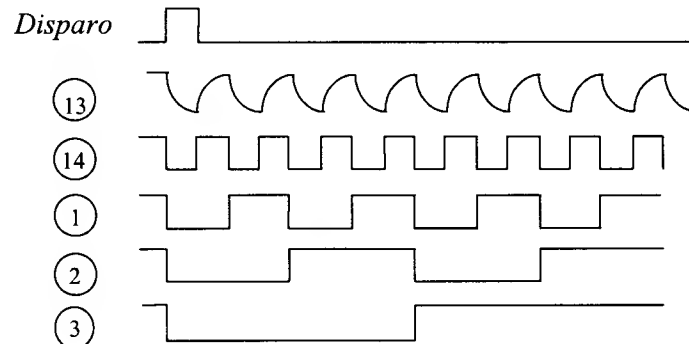
Por otro lado, el oscilador interno puede inhibirse para usar una base de tiempos externa que convierte al circuito en un contador programable y un generador de ocho suboctavas de la frecuencia base. El cronograma de la *figura 10.17.b* muestra las formas de onda correspondientes a la conexión generalizada. El ciclo comienza con la aplicación de un pulso de disparo que facilita al contador, pone todas sus salidas a baja e inicializa la base de tiempos. La capacidad externa se carga a través de  $R$  desde  $0,2 V$  a  $0,7 V$  generando una oscilación de periodo  $T = RC$ . En las descargas se generan los pulsos que pasan al contador y el ciclo termina cuando se genera un pulso de puesta a cero (*reset*). Este pulso inhibe a la base de tiempos y al contador, pasando a alta todas las salidas del contador, incluida la de arrastre.

Las salidas del contador son transistores en colector abierto que permiten la programación mediante su conexión a un potencial positivo usando conmutadores manuales o por control electrónico

de conmutadores analógicos. Si todas las salidas están conectadas juntas (AND cableado), la función de salida estará en baja hasta el final pero el ciclo puede acabar antes, con sólo conectar las salidas adecuadas del contador a su entrada de *reset* de la misma forma que vimos en el tema de contadores para números que no eran múltiplos de 2. Si no se conecta a *reset* ninguna de las salidas el circuito opera como un oscilador a partir de la aplicación del pulso de inicio.



(a)



(b)

**Figura 10.17.** Ejemplo de temporizador programable. (a) Esquema del circuito interno. (b) Cronograma mostrando la forma de las señales en algunos puntos.

Este tipo de circuito es representativo de la tendencia en el desarrollo de circuitos temporizadores potentes que, por un lado, camina hacia el desarrollo de controladores específicos (por ejemplo para cada tipo de microprocesador) y por otro lado, hacia temporizadores programables de uso general que no sólo generan las funciones elementales del modo estable y monoestable, sino que permiten sintetizar trenes de pulsos de forma prefijada.

## 10.6. RELOJES

Los relojes son circuitos de tiempo en la línea de las soluciones programables expuestas en el apartado anterior pero de mucha mayor precisión y estabilidad en la frecuencia de oscilación. La clave de su diseño está en usar un cristal de cuarzo para definir la frecuencia de oscilación, en sustitución de la red R-C pasiva. El resto del reloj incluye amplificadores, circuitos conformadores de onda, contadores para dividir la frecuencia y un conjunto de señales de control adicionales para facilitar o inhibir, sincronizar o establecer condiciones iniciales (*reset*). Cuando el reloj genera más de una salida se llama polifásico y la forma de onda en cada una de las fases mantiene una relación precisa con las otras fases, necesaria, por ejemplo, para controlar la inyección, el transporte y la extracción de la carga en memorias dinámicas.

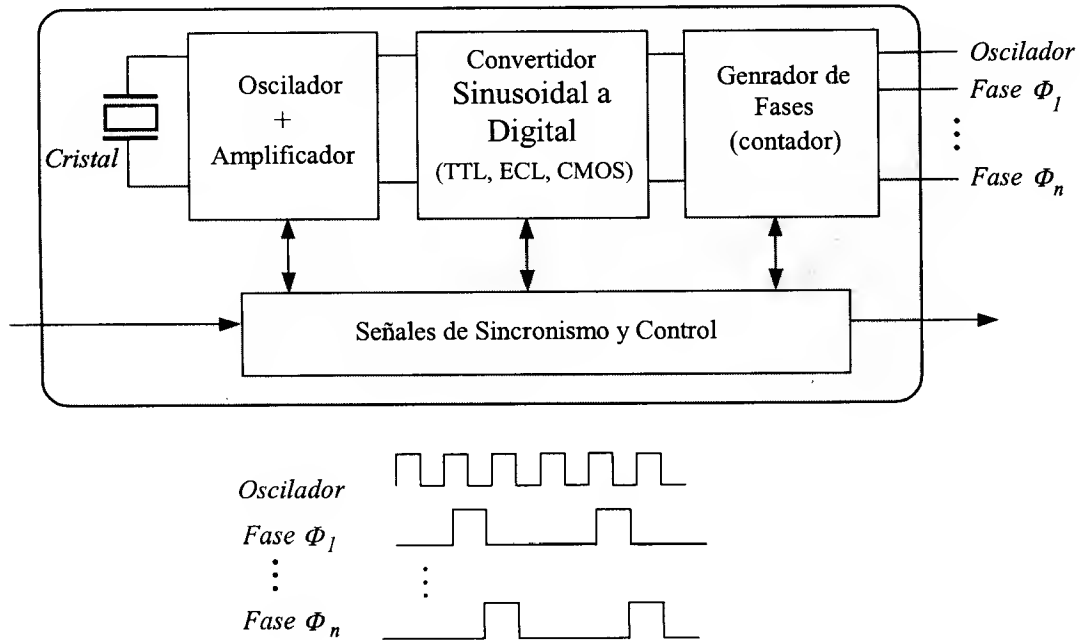
El esquema general de un circuito de reloj es del tipo mostrado en la *figura 10.18*. Consta de un oscilador, un amplificador, un circuito de paso de señal sinusoidal a señal digital con niveles de una o más familias lógicas (ECL, TTL o CMOS,) y un circuito generador de las distintas fases que incluye, en general, un contador y un conjunto de puertas lógicas y retardos. Todas estas funciones están usualmente integradas de forma que para el uso del reloj como elemento de circuito sólo nos hace falta añadir la alimentación y el cristal de cuarzo piezoeléctrico que fija la frecuencia de oscilación debido a que su impedancia es la propia de un circuito sintonizado de alta precisión tal como veremos a continuación.

Los osciladores a cristal son osciladores sintonizados en los que el circuito resonante ( $R, L, C$ ) se sustituye por un cristal. La *figura 10.19* muestra el principio de funcionamiento de este tipo de oscilador. En un circuito L-C hay frecuencias bajas en las que domina la impedancia de la rama capacitiva ( $X_C = 1/C\omega$ ). Para frecuencias altas domina la impedancia de la rama inductiva ( $X_L = L\omega$ ) y existe una frecuencia, llamada frecuencia de resonancia en paralelo, en la que las impedancias de ambas ramas se compensan dando lugar a una impedancia nula. Se cumple entonces que  $X_L = X_C$  y la frecuencia a la que esto ocurre,  $\omega_0$ , se llama *frecuencia de resonancia* y se calcula igualando la



expresión de la reactancia inductiva con la correspondiente a la reactancia capacitiva de forma tal que:

$$X_L(\omega_o) = X_C(\omega_o) \Rightarrow L\omega_o = \frac{1}{C\omega_o} \Rightarrow \omega_o = \frac{1}{\sqrt{LC}} \quad [10.20]$$



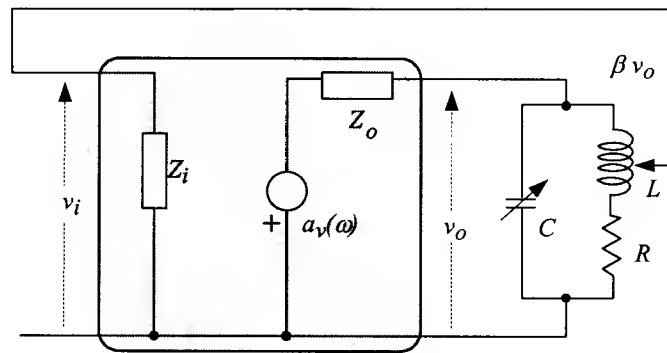
**Figura 10.18.** Esquema general de un reloj polifásico.

Esta situación ideal se ve alterada en la realidad debido a las pérdidas en la bobina, lo que exige considerar una pequeña resistencia,  $R$ , en serie con  $L$ . Entonces, para que el circuito oscile es necesario cerrar el lazo de realimentación a través de un amplificador que compense las pérdidas de forma que la ganancia total del lazo sea la unidad para esa frecuencia,

$$A_v(\omega_0) \cdot [ \text{Pérdidas en } (R, L, C) ] = 1$$

Cuando queremos que la oscilación sea precisa y muy estable, usamos un cristal de cuarzo como circuito resonante para simular la red  $(R, L, C)$ . Cuando a un cristal de cuarzo piezoeléctrico se le aplica un potencial eléctrico externo sufre deformaciones que dependen de forma estricta de su estructura y se comporta como un circuito resonante de acuerdo con el esquema equivalente de la

figura 10.20.a con dos posibilidades de resonancia: *serie* (frecuencia  $\omega_s$ ) en la que la impedancia es cero y *paralelo* (frecuencia  $\omega_p$ ) en la que la impedancia es infinito. Ambas frecuencias ( $\omega_s$  y  $\omega_p$ ) están muy próximas y la pendiente de la curva de impedancia es muy alta, de forma que si introducimos al cristal en un circuito que le compense de las pérdidas, este oscilará con una frecuencia fija extremadamente precisa y estable. La figura 10.20.b muestra un ejemplo de posible configuración en la que el cristal se incluye en el lazo de realimentación positiva de un amplificador operacional. En electrónica digital, el amplificador se sintetiza usando la misma tecnología que las puertas lógicas. Por ejemplo, mediante un par diferencial en ECL o usando la zona de transición de un inversor CMOS, tal como comentamos en el tema correspondiente.



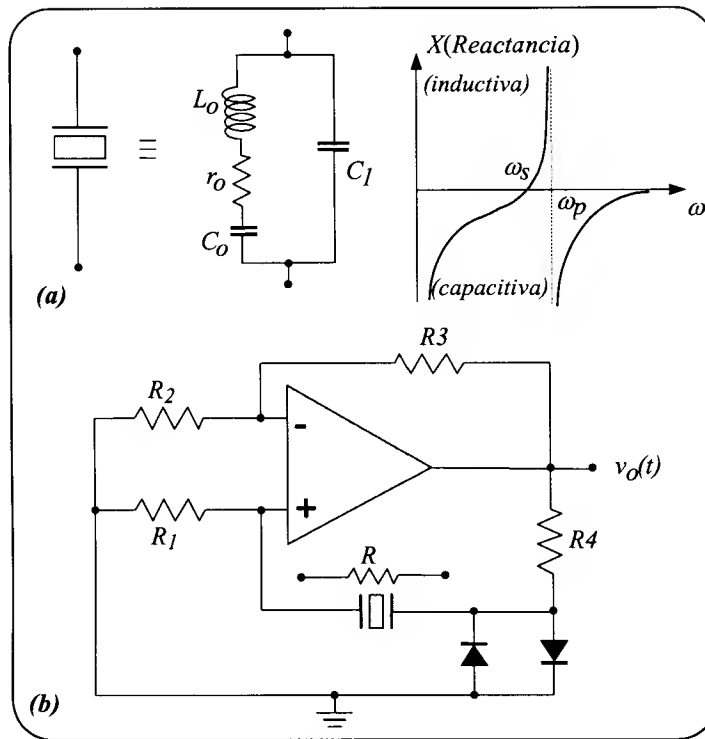
**Figura 10.19.** Esquema general de un oscilador sintonizado. Consta de un amplificador con una impedancia de carga  $RLC$  que fija la frecuencia de oscilación.

En el ejemplo de la figura el cristal opera en modo serie de modo que a la frecuencia de resonancia se comporta como una resistencia de bajo valor ( $R$ ). El amplificador opera en configuración no inversora con una ganancia,

$$A_v = \left( 1 + \frac{R_3}{R_2} \right) \quad [10.21]$$

Como la entrada es una parte de la salida realimentada a través del partidor de tensión ( $R_1$ ,  $R_4$ ,  $R_{del\ cristal}$ ), la condición de oscilación será:

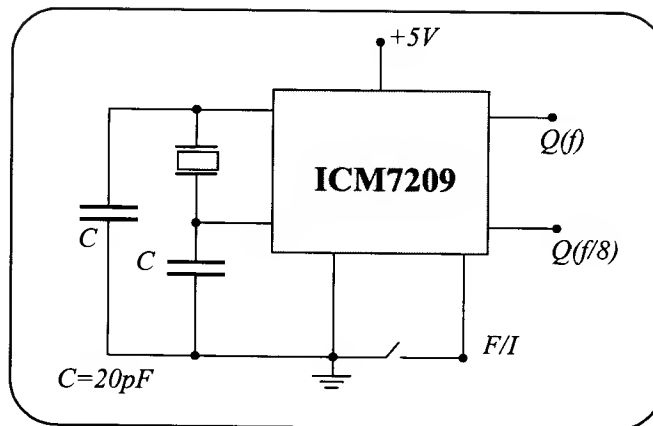
$$\left( \frac{R_1}{R_1 + R_4 + R} \right) \cdot \left( 1 + \frac{R_3}{R_2} \right) \geq 1 \quad [10.22]$$



**Figura 10.20.** Osciladores sintonizados usando cristales de cuarzo para fijar la frecuencia de oscilación.  
**(a)** Circuito equivalente del cristal y representación del valor de la reactancia con la frecuencia.  
**(b)** Síntesis mediante un amplificador operacional.

Finalmente, la *figura 10.21* muestra el uso de un cristal en un oscilador integrado en tecnología CMOS (ICM7209), representativo de los niveles más sencillos de relojes monofásicos. El circuito incluye el amplificador-conformador de onda, un contador que divide por 8 y una entrada de *facilitación/inhibición* (F/I). En la misma línea está el circuito MC12061 con tecnología ECL y salidas ECL y TTL.

Existe una amplia variedad y especificidad en los circuitos de reloj. De hecho, cada microprocesador posee un circuito de reloj específico que, con sólo conectarle el cristal de cuarzo externo, genera todas las señales que hacen falta para sincronizar y controlar el tráfico de datos e instrucciones dentro del microprocesador.



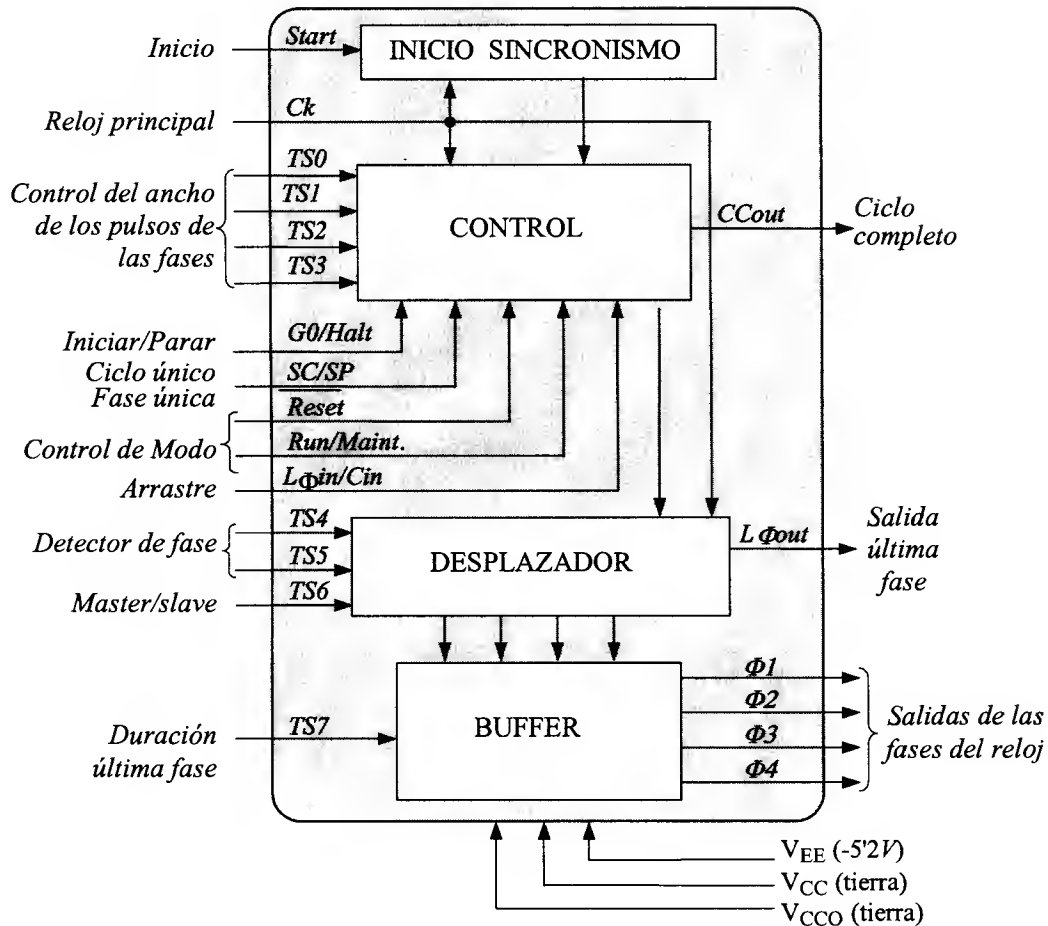
**Figura 10.21.** Ejemplo de reloj integrado tipo ICM7209. Para su uso sólo necesitamos conectar externamente la alimentación y el cristal de cuarzo. El resto de las funciones de amplificación y conformación de ondas ya las incluye el circuito.

La *figura 10.22* resume el diagrama de bloques funcionales y el significado de los terminales correspondientes al reloj MC10802 en tecnología ECL 10K de Motorola. Hemos elegido este circuito por su carácter de bloque funcional general en LSI de alta complejidad y versatilidad, sin ser específico de ningún controlador. Su diagrama de bloques responde de forma muy próxima al esquema general de reloj polifásico propuesto en la *figura 10.16*, salvo que su punto de partida es un reloj monofásico. El circuito incluye la lógica y las líneas de control necesarias para generar las cuatro fases del reloj y facilitar las operaciones de inicio, parada y diagnóstico. El circuito admite la expansión serie si fueran necesarias más de cuatro fases y sus principales características son la programabilidad del número de fases, inicio de oscilación sincronizable y reset asíncrono.

La programación de las fases se realiza a través de las líneas *TS4* y *TS5*. Cuando las dos están en alta hay una fase, cuando *TS5* está en alta y *TS4* en baja, hay dos fases, etc. Cada fase se considera activa cuando está en baja y su duración puede controlarse también a través de *TS0*, *TS1*, *TS2* y *TS3*. El final de cada fase es una transición de baja a alta, tal como se requiere en los relojes que activan a los biestables "master-slave" en la familia lógica ECL.

La selección de una fase ( $TS4=1$ ,  $TS5=1$ ) se usa sólo con circuitos en cascada, por ejemplo para obtener un reloj con cinco fases a partir de dos circuitos MC10802. Si sólo usamos un circuito, la función de reloj monofásico se obtiene programándolo como bifásico de forma que  $\Phi_1$  es la fase

deseada y  $\Phi_2$  su complementaria.



**Figura 10.22.** Ejemplo de reloj polifásico en ECL (MC10802). Esquema funcional y terminales.

Las líneas de *control* determinan el modo de operación del reloj (inicio, parada o diagnóstico). Hay cuatro líneas de control:  $\overline{RESET}$ , Run/Maint (*R/M*), Co/Halt (*G/H*) y Single Cycle/Single Phase (*SC/SP*). El  $\overline{RESET}$  es una línea asíncrona que domina a las otras tres. Cuando está activa (en baja) pasa a todas las fases a un estado de alta y se usa como parte del procedimiento de inicialización.

El *modo de mantenimiento* se selecciona con *R/M* en alta y posee a su vez cuatro configuraciones de trabajo controladas por las otras dos líneas (*G/H* y *SC/SP*). Cada operación en modo de mantenimiento se inicializa con las transiciones baja a alta de la línea *START*. Las cuatro

configuraciones son:

<i>R/M</i>	<i>G/H</i>	<i>SC/SP</i>	<i>Función</i>
H	L	L	a) <i>Parar al final de un ciclo</i>
H	L	H	b) <i>Parar al final de una fase</i>
H	H	L	c) <i>Avanza un único ciclo</i>
H	H	H	d) <i>Avanza una única fase</i>

En las dos primeras configuraciones (a y b), el funcionamiento del reloj es continuo. En el caso a), con *G/H* y *SC/SP* en baja, la oscilación comienza con *Start* y continúa hasta terminar el ciclo durante el cual la entrada de control *G/H* ha pasado a alta, como se muestra en la primera parte del cronograma de la *figura 10.23*. En el caso b), el funcionamiento también es continuo y se controla con *G/H* en baja y *SC/SP* en alta, pero ahora al recibir la señal de parada (*G/H* pasa a alta) el circuito se para al final de la fase que en ese momento está activa, primera parte del segundo cronograma de la *figura 10.23*.

Los modos c) y d) son de ciclo único o de fase única, respectivamente. El primero es una función de diagnóstico seleccionada con *G/H* en alta y *SC/SP* en baja y se usa para avanzar sólo un ciclo a partir de la posición en la que actúa el control *Start*, es decir, cada vez que hay un pulso de *Start* pasa sólo un ciclo completo, es decir, las cuatro fases. (segunda parte del primer cronograma de la *figura 10.23*). En la configuración d) el funcionamiento es el mismo pero ahora dejando pasar una fase cada vez que se presenta el pulso de *Start* y *G/H* y *SC/SP* están en alta (segunda parte del segundo cronograma). La fase que pasa con cada pulso de *Start* es la fase siguiente a la que paso con el anterior pulso de *Start*.

El segundo modo general de funcionamiento del reloj (*Run*) se selecciona con *R/M* en baja y las cuatro configuraciones del modo descrito anteriormente sólo se diferencian en la fuente de la que procede la señal. Ahora estas operaciones continuas se inician con *G/H* en baja y no necesitan una señal de *Start* adicional. La señal de salida de ciclos completos (*CC<sub>out</sub>*) monitoriza el estado funcional del reloj indicando cuando está funcionando o en la condición inicial.

Las formas de onda mostradas en el cronograma de la *figura 10.23* corresponden al *modo de mantenimiento* y son representativas de la estructura general de todos los relojes polifásicos. Incluye el oscilador principal (*Ck*) que marca la máxima frecuencia en el circuito, las cuatro fases de menor frecuencia y desplazadas, las señales de control asíncrono (*Reset*, *SC/SP*, *Start* y *G<sub>0</sub>/Halt*) y el *bit de*

status ( $CC_{out}$ ), que monitoriza el estado del reloj.

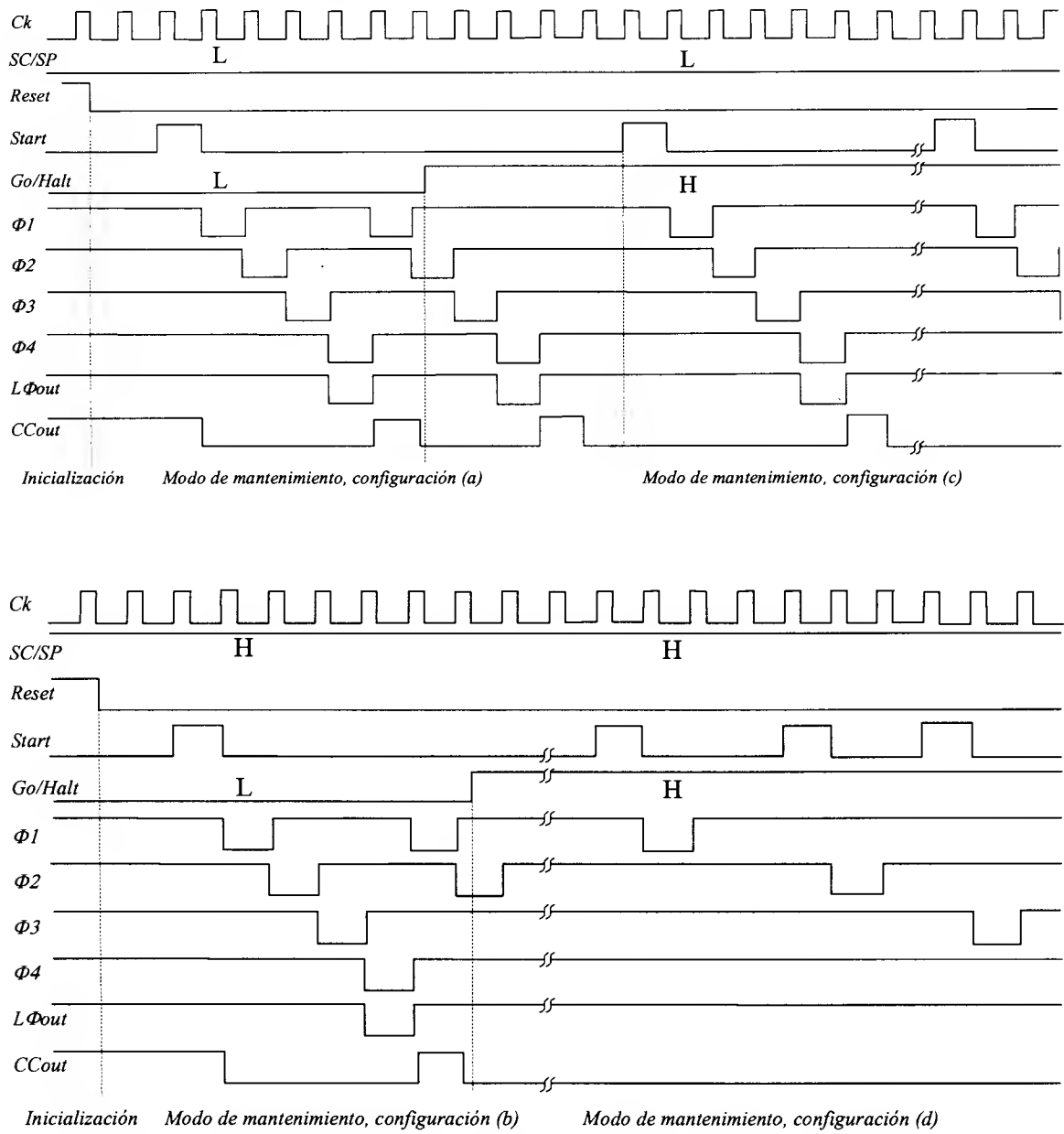


Figura 10.23. Cronograma del reloj polifásico MC10802 cuando está operando en modo de mantenimiento.

## 10.7 PROBLEMAS

En este capítulo hemos estudiado un conjunto de circuitos cuya misión es generar secuencias de pulsos de separación y/o duración variable para controlar el paso coordinado de señales entre distintos puntos de un sistema digital. Hemos visto que los elementos básicos son el oscilador y el monoestable. El primero genera una secuencia periódica de pulsos en general de corta duración y en forma de onda cuadrada. El segundo genera pulsos aislados de inicio y fin controlables. Con la combinación de ambos, junto con algunos elementos de lógica combinacional y secuencial de los que hemos estudiado en temas anteriores, se pueden obtener todos los circuitos necesarios para temporización digital.

Hemos dedicado un apartado final al estudio de los relojes monofásicos y polifásicos. Esencialmente, son osciladores de *precisión* controlados por cristales de cuarzo piezoeléctrico, que generan una señal periódica muy precisa. Poseen además, circuitos de control y divisores de frecuencia (contadores) que permiten controlar varias fases.

Así pues, el objetivo de los ejercicios de este capítulo es ayudar al alumno a manejar los circuitos de tiempo en los dos tipos de tarea usuales en electrónica digital: *análisis* y *diseño*. Es decir, podemos partir de un circuito temporizador concreto y preguntarnos por cuál es la forma de onda que genera ese circuito en tal o cual punto de interés. Inversamente, existirán ejercicios correspondientes a necesidades de diseño en el desarrollo de aplicaciones reales, en los que partiremos de una forma de onda y pediremos el diseño de un circuito que la genere.

En ambos tipos de ejercicio (*análisis* y *síntesis*) se parte del supuesto que los módulos de diseño son circuitos integrados tipo 121, 123 ó 555, por ejemplo. Existen sin embargo ejercicios complementarios en los que el propósito es reforzar la comprensión del funcionamiento interno de esos circuitos integrados. Ambos tipos de ejercicios son necesarios para alcanzar una visión razonablemente amplia e integral de la electrónica digital.

**E.10.1.** *El circuito de la figura incluye un monoestable sintetizado con puertas del tipo de la figura 10.3, junto con un generador de los pulsos de disparo. Dibujar las formas de onda en los puntos (1), (3) y (5) como respuesta al tren de pulsos que se muestra en la primera línea del cronograma. Suponiendo que las puertas usadas son TTL y que la alimentación es  $V_{CC}=5\text{ V}$ . ¿Cuál es la expresión analítica de la forma de onda en el punto (3), que mide la variación de la carga en el condensador?*





se muestra en las figuras 10.6 y 10.7. ¿Cómo realizarías las conexiones para, usando dos circuitos de este tipo conseguir:

- (a) Un circuito astable de frecuencia de oscilación 200kHz y onda cuadrada.
- (b) Un circuito astable de frecuencia de oscilación 100kHz y onda rectangular con duración del pulso,  $t=2 \cdot 10^{-6}$  s.

Dado que esto es un circuito comercial el cálculo de los valores de los componentes externos ( $R_{ext}$  y  $C_{ext}$ ) hay que hacerlo de acuerdo con las especificaciones del fabricante, que para este circuito son:

- 1) Para  $C_{ext} \leq 10^3$  pF el fabricante da una familia de curvas en las que se representa la anchura del pulso frente a la  $C_{ext}$  y como parámetro  $R_{ext}$ .
- 2) Para  $C_{ext} > 10^3$  pF, la anchura del pulso viene dada por

$$t_w = K R_{ext} C_{ext} \left( 1 + \frac{0,7}{R_{ext}} \right)$$

donde  $K=0,32$  para el circuito SN74122 (simple) y  $K=0,28$  para el circuito SN74123 (doble). En ambos casos, en esta expresión se considera que el valor de  $R$  viene dado en  $k\Omega$  y el de  $C$  en pF. Se obtiene así la duración del pulso en ns. En nuestro caso,  $K=0.32$ .

**E.10.4.** El circuito de la figura 10.10 muestra el circuito interno del 555 y el esquema externo de conexión para configurarlo como astable o como monoestable. Rellenar los valores de tensión, estados lógicos y formas de onda de la siguiente tabla para  $V_{CC}=5$  V en los siguientes casos:

- (a) Configuración monoestable.
- (b) Configuración astable.

Reset	Disparo (2)	Umbral (6)	R	S	G	Vdrenador (7)	Extremos de C	Salida (3)
0	?	?	?	?	?	?	?	?
1	"	"	"	"	"	"	"	"
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.

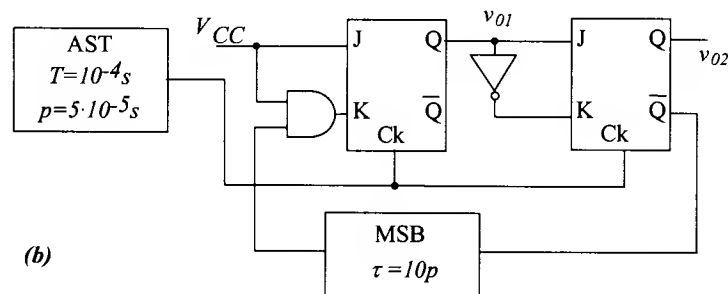
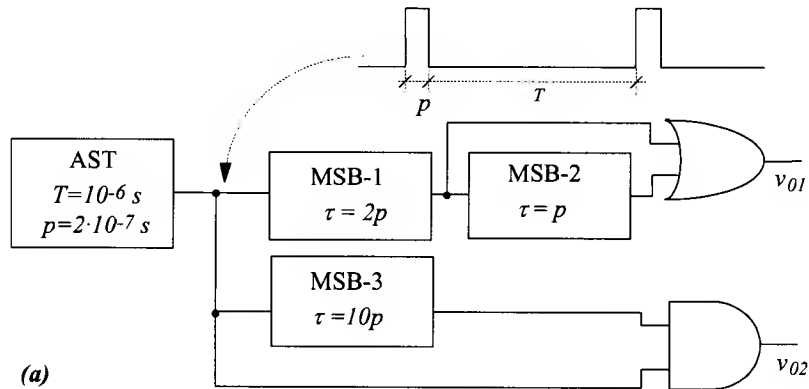
**E.10.5.** ¿Cómo podríamos sintetizar un divisor por 2 de la frecuencia de un tren de pulsos usando

un astable y un monoestable?. Dibujar el circuito y estimar valores para las  $R$  y  $C$  externas suponiendo que la frecuencia del tren de pulsos que debe generar el astable es  $f=1,2\cdot 10^6\text{Hz}$ . ¿Cómo podríamos modificar el circuito anterior para dividir por 3 la frecuencia inicial?.

**E.10.6.** En algún punto del capítulo hemos comentado cómo podría usarse un 555 como circuito detector de la omisión de un pulso en un tren de impulsos estableciendo un tiempo de retardo algo mayor que el intervalo entre dos pulsos sucesivos.

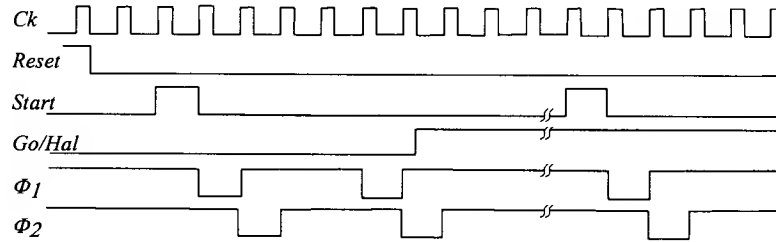
- Diseñar de forma precisa el circuito que realiza esta función para trenes de impulsos cuya frecuencia es  $10\text{ kHz}$  y la duración del pulso es de  $0,3\cdot 10^{-4}\text{ s}$ .
- ¿Qué modificaciones tendríamos que introducir en el apartado anterior para que sólo se detectara un fallo cuando faltaran dos pulsos sucesivos?.

**E.10.7.** Obtener las formas de onda que generan los circuitos de las siguientes figuras en  $v_{01}$  y  $v_{02}$ .



**E.10.8.** Diseñar un circuito que genere las formas de onda de la figura. Obsérvese que

corresponden de forma cualitativa a las de un reloj polifásico análogo al descrito en la figura 10.21, con sólo dos fases que se activan a partir de un pulso de "start" y siguen activas hasta el cambio de nivel de la señal de "halt". Para la síntesis se pueden usar monoestables, astables, biestables y puertas lógicas. Para fijar valores numéricos en las R y C externas, suponer que el reloj principal es de 1 MHz y que todos los pulsos duran  $0,5 \cdot 10^{-6}$  s.



---

# ***Preparación de la Evaluación***

---

El comentario final del apartado sobre guía de estudio vale también ahora para la autoevaluación. La mayor parte de los ejercicios que proponemos aquí tienen que ver con reflexiones, modificaciones y ejemplos análogos a los usados en el tema 10 del texto de problemas. Aquí hemos añadido, esencialmente, ejemplos de uso del simulador para poner de manifiesto cómo se puede modificar el comportamiento externo de un temporizador actuando sobre su estructura interna y/o sobre sus componentes externas de programación. Fianalmente, la redacción está orientada a la preparación de exámenes. Los enunciados se parecen de hecho a preguntas propuestas en cursos anteriores.

## **Objetivos**

- 1: *Circuitos monoestables y astables*
- 2: *Circuitos tipo 555 y temporizadores programables*
- 3: *Relojes monofásicos y polifásicos*

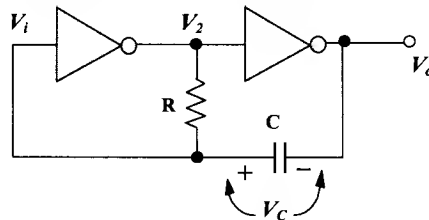
## **1: Circuitos monoestables y astables**

- 1.1. *Describa el comportamiento externo de un monoestable. ¿Qué señales externas necesita?. ¿Sobre qué parte de la señal de salida tenemos control?.*
- 1.2. *En la figura 10.3 de este tema describimos el diseño de un monoestable a partir de la realimentación de un inversor y una puerta NOR acoplados a través de una red R-C. Por otro lado recuerde que en el tema 8 (figura 8.10) estudiamos la realimentación positiva de dos inversores como base de la biestabilidad. Reflexione sobre las analogías y diferencias entre ambas arquitecturas aparentemente idénticas pero que en un caso conducen a dos estados estables y en el otro a un estado estable y otro metaestable. Explique la diferencia.*
- 1.3. *Estudie conceptual y analíticamente la forma de onda que aparece en los puntos 2, 3 y 4 de la figura. Dibuje la evolución temporal de las señales en esos puntos cuando a la entrada se aplica un pulso positivo. ¿Qué pasa al duplicar el valor de C?.*
- 1.5. *Describir el comportamiento externo del astable y volver a razonar sobre el hecho que una*

misma arquitectura (dos inversores cerrando un lazo de realimentación positiva) haya dado lugar a 1) circuitos biestables, 2) circuitos monoestables y 3) circuitos astables. Explique cómo la red pasiva de acoplo (R, C, R-C) se encarga de marcar la diferencia. Obsérvese como las memorias “digitales” en el fondo son analógicas porque dependen de un proceso continuo de carga y descarga de una capacidad. Piense también en que lo que las hace digitales es la existencia de mecanismos no lineales que “cortan” esos procesos de carga y descarga al llegar a cierto valor umbral.

- 1.6. Recuerde que el principio de funcionamiento de un astable es la detección de un potencial, de valor preestablecido, en los extremos de un condensador. Este potencial se presenta tras un cierto tiempo definido por la constante de tiempo de carga y descarga del condensador ( $\tau=RC$ ), y por el valor de la fuente de alimentación,  $V_{CC}$ , hacia la que tiende la tensión en bornas del condensador. Al alcanzar este valor hay un cambio brusco en el estado de los inversores y comienza de nuevo todo el ciclo de carga/descarga.

Para puertas en CMOS la tensión de disparo es  $V_{CC}/2$ . Compruebe entonces las formas de onda de la figura, obtener su expresión analítica y demostrar que  $T=2R \cdot C \cdot \ln 2$ .



Si tiene dificultades en la solución de este ejercicio le recomendamos que revise el ejercicio E.10.2 del texto de problemas.

## 2: Circuitos tipo 555 y Temporizadores Programables

En el resumen se muestra la estructura interna del 555 y las posibilidades de programarlo externamente para que actúe como monoestable o como astable. Sobre este circuito vamos a desarrollar diferentes ejercicios de autoevaluación

- 2.1. Describa el significado funcional de cada uno de sus terminales en términos del papel que juega en la estructura interna. Por ejemplo, “el terminal 4 actúa de reset porque entra a través de un inversor a la puerta NOR de reset de un “latch” R-S de forma que .....”. O bien “el

terminal 6 cumple el rol del umbral superior de disparo por entrar en el terminal no inversor de un comparador diferencial cuya entrada inversora está fija a 2/3 de la tensión de alimentación. Entonces cuando la tensión en él supera ....". etc.

**2.2.** Conecte la parte externa del circuito para que opere como monoestable. Explique su funcionamiento y compruebe la validez de la tabla que se incluye en el resumen. Si encuentra dificultades, revise el apartado 10.4 del texto de teoría y el ejercicio E.10.4 del texto de problemas.

**2.3.** Repita el proceso de análisis para la configuración astable.

**a)** Conecte adecuadamente los conmutadores externos y explique el funcionamiento

**b)** Trate de construir una tabla análoga a la del ejercicio anterior.

Si encuentra dificultades, revise el apartado 10.4 del texto de teoría y el ejercicio E.10.4 del texto de problemas.

Supongamos que al llegar aquí ya conocemos bien cómo funcionan los astables y los monoestables, de forma que ya podemos usarlos como elementos de circuito para realizar otras funciones.

Recordemos también que la duración del estado metaestable en el monoestable es:

$$t_1 = R_A C \ln 3$$

y que la frecuencia del astable es:

$$f = \frac{1}{T} = \frac{1}{t_1 + t_2} = \frac{1}{(R_A + 2R_B) C \ln 2}$$

Intente resolver los siguientes ejercicios:

**2.4.** Diseñar un monoestable en el que la duración del estado metaestable varíe entre  $17\mu\text{s}$  y  $8517\mu\text{s}$ .

**a)** Fije un valor para C y calcule los valores de R

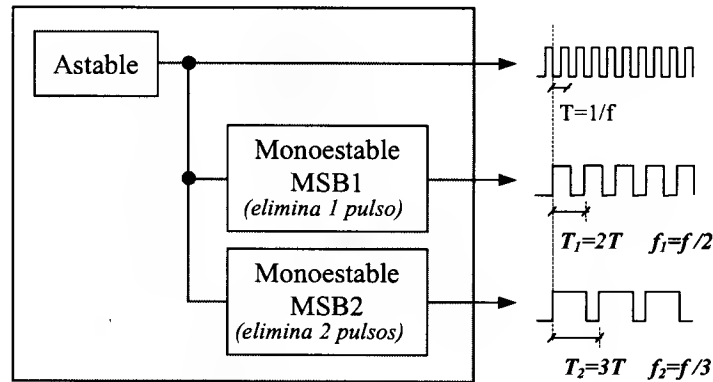
**b)** Fije un valor de R y calcule los valores de C

¿Cual de los dos métodos es más conveniente y por qué?

**2.5.** Diseñar un astable que genere pulsos de  $50\mu\text{s}$ . de anchura y frecuencia variable entre 5kHz y 10KHz.

- 2.6. Conectar un astable y 2 monoestables para que el circuito resultante actúe como divisor por 2 y por 3 de la frecuencia principal.

Calcular los valores de las  $R$  y las  $C$  correspondientes del astable y de los monoestables para que la frecuencia principal sea de 1MHz.



- 2.7. En el tema hemos comentado cómo podría usarse un 555 como circuito detector de la omisión de un pulso en un tren de impulsos estableciendo un tiempo de retardo algo mayor que el intervalo entre dos pulsos sucesivos.

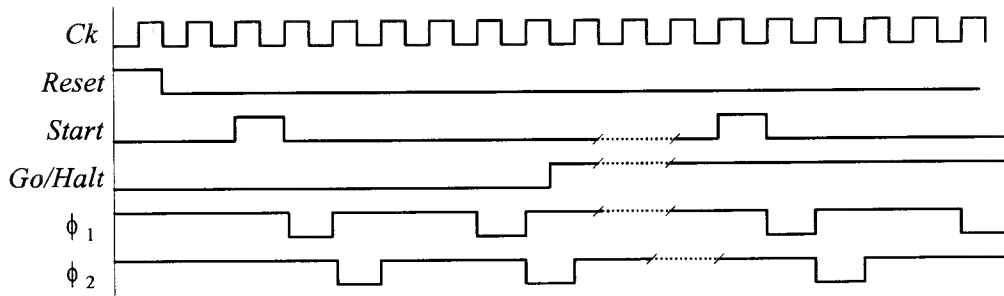
- Diseñar de forma precisa el circuito que realiza esta función para trenes de pulsos cuya frecuencia es 10 kHz y la duración del pulso es de  $0,3 \cdot 10^{-4}$  s.
- ¿Qué modificaciones tendríamos que introducir en el apartado anterior para que sólo se detectara un fallo cuando faltaran dos pulsos sucesivos?

Si encuentra dificultades al resolverlo puede consultar el problema E.10.6.

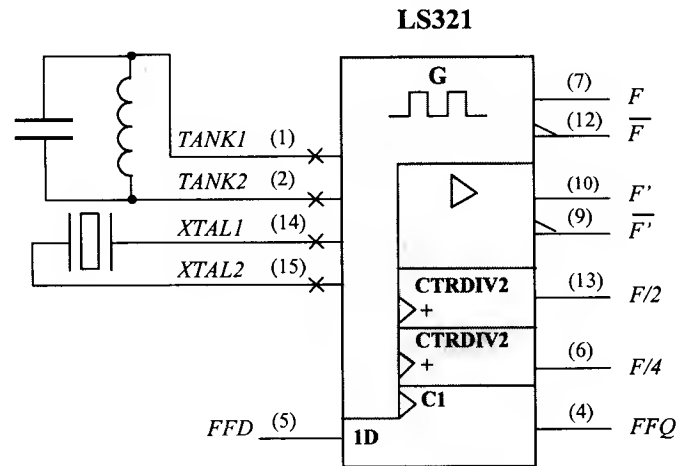
### 3: Relojes Monofásicos y Polifásicos

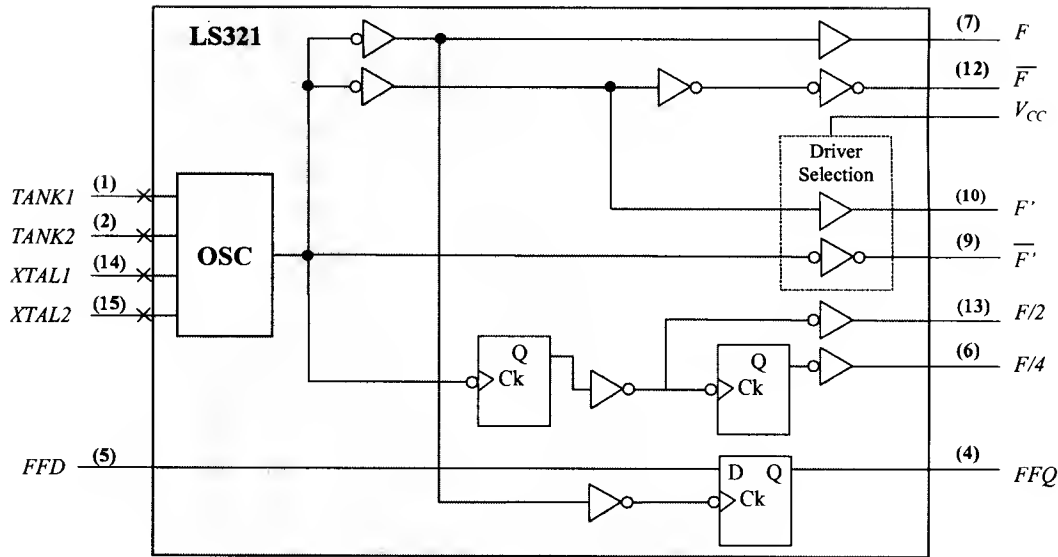
- 3.1. Diseñar un circuito que genere las formas de onda de la figura. Obsérvese que corresponden de forma cualitativa a las de un reloj polifásico, con sólo dos fases que se activan a partir de un pulso de start y siguen activas hasta el cambio de nivel de la señal de Halt. Para la síntesis se pueden usar monoestables, astables, biestables y puertas lógicas. Para fijar valores numéricos en las  $R$  y  $C$  externas, suponer que el reloj principal es de 1 MHz. y que todos los pulsos duran  $0,5 \mu$  s.





- 3.2. La mayoría de los osciladores controlados por cristal que van a ser usados como relojes de un sistema digital integran todas las funciones (incluido el amplificador de alta ganancia) salvo el propio cristal. En el tema hemos visto la solución en CMOS (ICM7209). Analice ahora la solución TTL, SN74321 cuyo circuito interno se muestra en la figura adjunta y que opera entre 1 MHz y 20 MHz e incluye salidas complementarias de frecuencia mitad y un cuarto. El cristal se conecta entre XTAL-1 y XTAL-2 y el circuito resonante paralelo entre TANK-1 y TANK-2 como se ha representado en la figura junto con su símbolo lógico.





Todos los procesadores actuales incorporan sus propios relojes polifásicos, en general de complejidad muy superior a las soluciones mostradas en este tema. Nuestro propósito ha sido más modesto. Sólo deseamos que a partir de ahora, en el estudio de memorias y en el resto de la docencia de la Electrónica Digital incluida en otras asignaturas, el alumno sepa cómo generar un pulso de duración controlable, cómo generar un tren de impulsos y cómo usar estos componentes básicos de la función de temporización para generar las estructuras temporales de control necesarias en un sistema de cálculo digital. Hemos visto que, esencialmente, esto se consigue con astables o relojes, monoestables, contadores binarios y algo de lógica combinacional adicional.

## REFERENCIAS BIBLIOGRÁFICAS

1. Grebene, A.B.: “Bipolar and MOS Analog Integrated Circuit Design”. J. Wiley & Sons. 1984
2. Texas Instruments. “*The TTL Data Book*”. 1987.
3. [www.fairchildsemi.com](http://www.fairchildsemi.com)
4. [www.national.com](http://www.national.com)



# TEMA

# 11

---

## ***Memorias RAM y CAM***

---

### **+ CONTEXTO**

*A lo largo de todo el texto hemos usado la estrategia de reflexionar primero sobre un conjunto de necesidades computacionales y estudiar después los circuitos electrónicos que soportan la satisfacción de esas necesidades, haciendo siempre referencia a los sistemas de cálculo digital.*

*Así hicimos al estudiar la lógica combinacional y las necesidades de circuitos que realizaran operaciones aritmético-lógicas y de rutas de datos y así volvimos a actuar al comienzo de los circuitos secuenciales. Vimos primero la necesidad de un nuevo modelo matemático y de circuitos biestables y después los estudiamos aplicados a la síntesis en general y al diseño de registros y contadores.*

*Con esto tenemos cubierta una parte importante de la electrónica digital que se necesita en arquitectura y tecnología de computadores. Sin embargo, si tuviéramos que decidir cuáles son las*

*funciones electrónicas más necesarias en computación en términos del área de silicio que ocupan, no cabe duda que esa función es la de almacenamiento transitorio o permanente de datos e instrucciones. Es decir, de las funciones de memoria en los distintos niveles (registros internos, caché, memoria principal, etc...). Además, cada vez se piden más prestaciones a las memorias por parte de los procesadores, por lo que este apartado de la electrónica digital es quizás el más dinámico ya que sigue habiendo una gran distancia entre lo que requiere un procesador a 800MHz y lo que ofrece una memoria a 250MHz.*

*En el tema 7, cuando estudiamos la lógica combinacional programable, hicimos una clasificación de las memorias distinguiendo primero entre las de “sólo lectura” y las de “lectura y escritura” y dentro de estas últimas entre no volátiles (EEPROMs) y volátiles. Aquí vamos a empezar el estudio de las volátiles. Es decir, de los circuitos de memoria activos, que usan biestables tipo R-S para almacenar la información..*

*Recordemos que allí terminábamos la clasificación de las memorias activas usando como último criterio la forma de direccionamiento (aleatorio, por contenidos y secuencial). Pues bien, en este tema 11 vamos a estudiar los dos primeros. Es decir, las memorias RAM estáticas (SRAM) y dinámicas (DRAM) y las memorias direccionables por contenido (CAM). En el próximo tema estudiaremos las memorias de acceso secuencial.*

*Obsérvese que bajo el nombre histórico de RAM (Random Access Memory), de lo que estamos hablando es de memorias de lectura-escritura (RWM), volátiles (que sólo mantienen información cuando está la fuente de alimentación), de acceso aleatorio (que cuesta el mismo tiempo acceder a cada una de las distintas direcciones) y que son estáticas, SRAM, porque almacenan la información en un biestable o dinámicas, DRAM, porque almacenan información en un condensador y necesitan mecanismos de “refresco” para reponer la pérdida de carga por corrientes de fugas.*

## **++ CONOCIMIENTO PREVIO NECESARIO**

*Para las funciones de organización, incluyendo la selección y el control de los procesos de lectura y escritura, necesitamos decodificadores y otros circuitos combinacionales y secuenciales que ya conocemos de temas anteriores. Quizá, para el estudio de los amplificadores sensores sea conveniente recordar algo de electrónica analógica referente al comportamiento de los amplificadores diferenciales que se usaron en el tema 4 para introducir la familia lógica ECL.*

*Finalmente, para el estudio de las distintas celdas SRAM y DRAM en bipolar y MOS no usaremos nada que no se pueda encontrar en los temas 2, 3 y 4. Básicamente, estas celdas son biestables R-S en los que se intenta refinar el diseño para conseguir que ocupen la menor área posible y, por consiguiente, se pueda conseguir la máxima densidad de integración.*

### **+++ OBJETIVOS DEL TEMA**

**Objetivo 1:** *Estudio de las memorias RAM estáticas (SRAM): Organización, celdas básicas en bipolar y MOS y ejemplos.*

**Objetivo 2:** *Estudio de las memorias RAM dinámicas (DRAM): Organización, celdas básicas y ejemplos.*

**Objetivo 3:** *Estudio de las memorias CAM: Organización, celdas básicas y ejemplos.*

*Al terminar el estudio del tema el alumno deberá tener una idea razonablemente clara y completa de la problemática electrónica asociada a las memorias activas, desde la búsqueda de la minimización de sus celdas básicas hasta los avances en organización y arquitectura encaminados a aumentar la velocidad y el ancho de banda (RAMs síncronas, modo “hiperpágina” o de salida de datos extendida, EDO, etc.)*

### **++++ GUÍA DE ESTUDIO**

*Hemos intentado presentar las memorias RAM y CAM sin una dependencia muy directa con circuitos concretos que sólo se usan como estrategia pedagógica ya que estamos convencidos de que sea cual fuere el circuito seleccionado va a ser sustituido por otro más veloz, de mayor capacidad y con mejores prestaciones en un corto intervalo de tiempo. Sin embargo, el descubrimiento de nuevas celdas básicas no parece tan probable y las necesidades de direccionamiento, control y circuitos de entrada/salida van a permanecer bastante invariantes. Por consiguiente, aconsejamos que se estudie el tema focalizando en los aspectos generales y comunes y que se acceda a través de la web a las distintas casas comerciales para conocer en cada momento el estado actual del mercado de las memorias RAM y CAM.*

*Finalmente, los ejemplos que incluimos aquí y los que se desarrollan en el texto de problemas*

*siempre usan **versiones mínimas** de cada tipo de memoria (“circuitos con dos palabras de dos bits”) para hacer manejable el estudio de los circuitos de direccionamiento y control de lectura/escritura así como la comprensión de los cronogramas correspondientes.*



---

# ***Contenido del Tema***

---

## **11.1. MEMORIAS DE LECTURA/ESCRITURA VOLÁTILES**

El esquema de la *figura 11.1* es una versión modificada del que vimos en el tema 7 para clasificar los distintos circuitos de memoria (*figura 7.8*). Ahora nos “olvidamos” de las PROM y las EEPROM y hacemos énfasis en las volátiles. Su característica distintiva es que usan circuitos activos (biestables) para almacenar la información. De hecho, los biestables del tema 8 y los contadores y registros del tema 9 son circuitos de memoria volátiles. Lo que justifica su estudio separado es el tamaño, el carácter masivo de la información que queremos almacenar primero y recuperar después de forma precisa y eficiente.

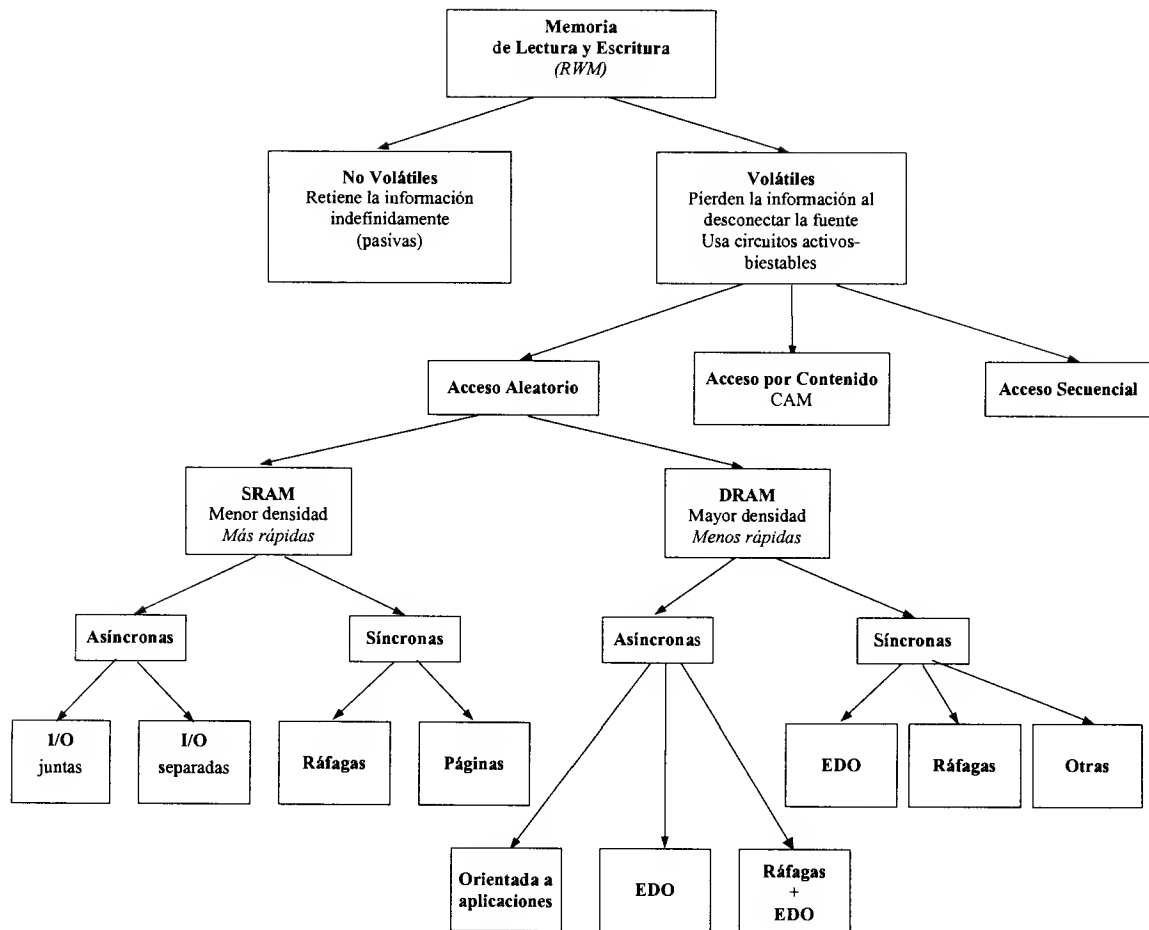
Dentro de las memorias volátiles y de acuerdo con el procedimiento de *direccionamiento*, es decir, de la forma de acceder a una posición concreta, tenemos:

- a) *Memorias de acceso aleatorio* (RAM)
- b) *Memorias de acceso por contenidos* (CAM)
- c) *Memorias de acceso secuencial* (FIFO, LIFO) (las estudiaremos en el tema siguiente).

En las memorias RAM, la información se organiza por bits o en palabras de  $n$  bits y estas palabras se almacenan en posiciones físicas que son accesibles de forma independiente. Es decir, existe la posibilidad de seleccionar una posición cualquiera, leer su contenido y/o escribir un nuevo dato. Estas memorias deberían llamarse de *lectura/escritura* y acceso *directo*. El tiempo de acceso a cada una de las distintas posiciones es el mismo, a diferencia de lo que ocurre con las memorias de acceso secuencial.

A su vez, las RAM se clasifican en estáticas (SRAM) y dinámicas (DRAM), atendiendo a la forma de almacenar la información (biestables o condensadores) y por consiguiente a las exigencias de circuitos periféricos adicionales para recuperar la pérdida de carga por corrientes de fuga.





**Figura 11.1.** Clasificación de las memorias de lectura/escritura.

En las memorias asociativas, o direccionables por contenido, CAM (Content Addressable Memory) se escribe de forma normal, análoga a las RAM, pero se lee por el contenido de las celdas de memoria, no por la dirección. Su proceso de lectura consiste en la comparación simultánea de un *contenido*, o parte de un contenido (bits no enmascarados) con todos los contenidos de la memoria o de un sector de la misma de forma que cuando existe una coincidencia se extrae la dirección correspondiente, o el resto de la información, caso de haberse realizado la comparación con sólo una parte de los bits.

Las memorias de acceso secuencial tienen la estructura de registros de desplazamiento, de

modo que se accede a su lectura y/o escritura de forma secuencial. Los datos pueden recircular en los registros o acumularse en estructuras FIFO (First-In, First-Out) o LIFO (Last-In, First-Out). En las primeras, el primer dato que entra es el que primero va a salir, cuando acabe de recorrer por desplazamiento la longitud del registro. En las segundas, el primer dato que entra es el último que sale porque los nuevos datos le empujan "hacia arriba", apilándose. Por eso a las organizaciones LIFO se les suele llamar también memorias "pila".

Finalmente, la selección de tecnología (bipolar, MOS, CMOS, BiCMOS, AsGa,...) está asociada a las necesidades de velocidad, consumo y densidad de empaquetamiento.

## 11.2. ORGANIZACIÓN DE LAS MEMORIAS SRAM

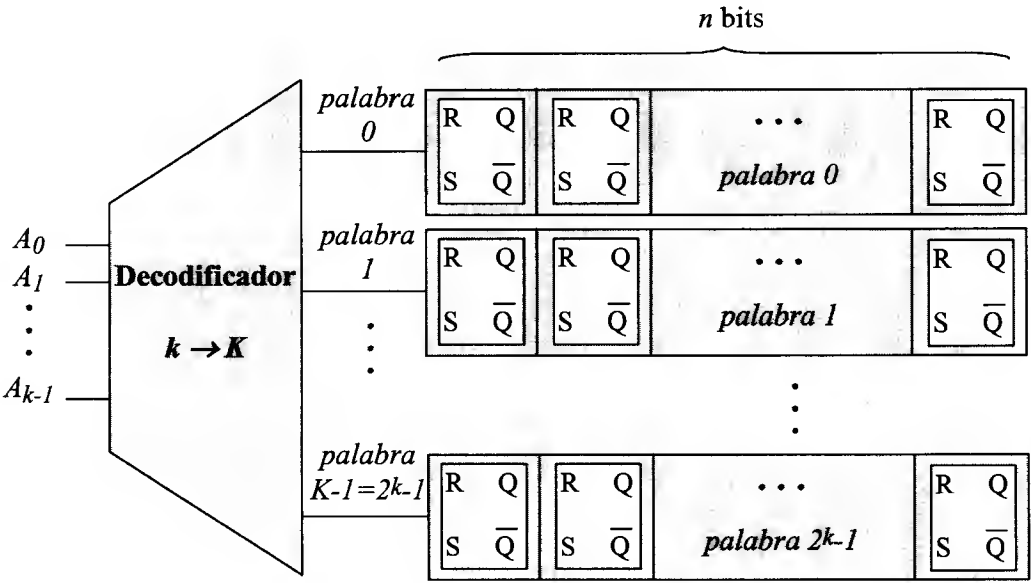
Las memorias RAM estáticas constan de un núcleo de almacenamiento formado por un número muy grande de biestables R-S y un conjunto de circuitos periféricos de direccionamiento y control de las operaciones de lectura/escritura.

El elemento básico, la celda R-S, almacena un bit y el conjunto de celdas (bits) se organizan en términos de bytes o palabras, que son grupos de  $n$  bits (8, 13, 32, ...) que se tratan globalmente. Cada dirección corresponde a una palabra. Así, con  $k$  bits podremos direccionar  $K=2^k$  palabras y la forma más inmediata de acceder a cada una de estas palabras es usando un decodificador de  $k$  a  $K$ , tal como se ilustra en la *figura 11.2*. Este decodificador sólo tiene activa una de sus  $2^k$  líneas de salida dependiendo de la dirección  $(A_0, A_1, \dots, A_{k-1})$  presente en la entrada. Al ser direccionada una palabra quedan accesibles para lectura/escritura todos sus bits.

Sin embargo, esta organización no es eficiente por el carácter longitudinal del circuito. Es más lógico buscar una estructura matricial de forma que las dimensiones horizontal y vertical del circuito sean comparables. Esto se consigue al repartir los bits de la dirección  $(A_0, A_1, \dots, A_{k-1})$  en dos grupos, uno para direccionar las filas de una matriz  $(A_l, A_{l+1}, \dots, A_{k-1})$  y el otro para direccionar las columnas  $(A_0, A_1, \dots, A_{l-1})$  de forma que el elemento direccionado es el correspondiente a la intersección de la fila (llamada *línea de palabra*) y la columna (llamada *línea de bit*) correspondientes a esos dos componentes de la dirección  $(A_l, A_{l+1}, \dots, A_{k-1})$  y  $(A_0, A_1, \dots, A_{l-1})$ .

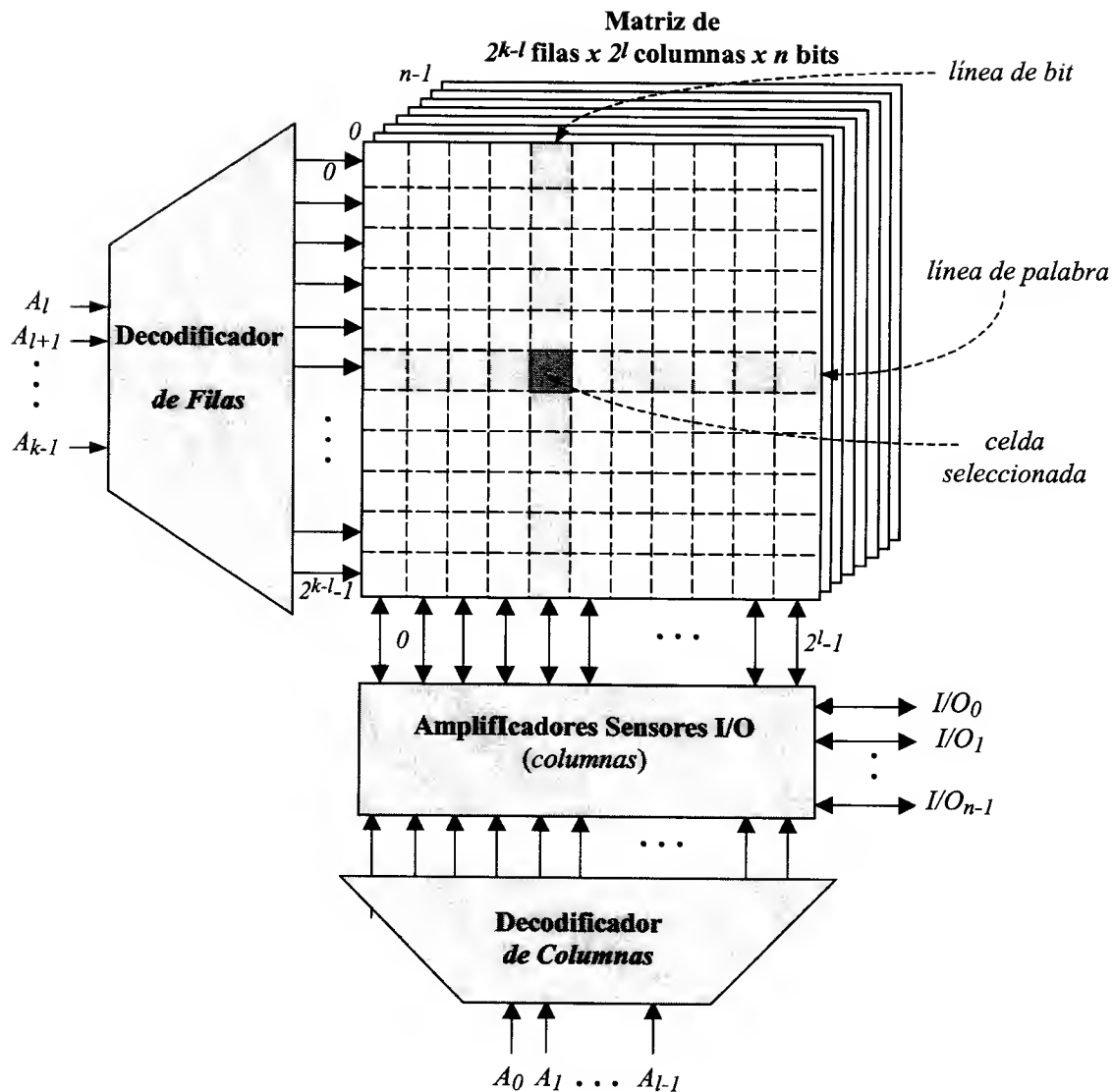
Si a estos circuitos decodificadores de la dirección y a la matriz de  $2^{k-l}$  filas (palabras) y  $2^l$  columnas (bits) le añadimos los circuitos de control de los procesos de lectura/escritura y de selección

de circuito y los circuitos de entrada y salida, obtenemos el esquema básico de organización de una memoria SRAM que se muestra en la *figura 11.3*. Obsérvese que lo que direccionamos con filas y columnas son las coordenadas de un plano. Es decir, una celda (un bit) por cada dirección. Si cada palabra tuviera 8 bits, por ejemplo, la imagen de la memoria sería tridimensional y podría conseguirse con uno o varios circuitos direccionados en paralelo. Por ejemplo, incluyendo los 8 bits en un único circuito, yuxtaponiendo dos circuitos de cuatro bits cada uno u ocho circuitos de un bit. Así, hablamos de circuitos como el INTEL2147H de 64x64x1 (64 filas, 64 columnas, 4096 palabras de un bit) o circuitos 256x128x8, es decir de 256 filas, 128 columnas, 32768 palabras de ocho bits).



**Figura 11.2.** Forma intuitiva de direccionar una memoria SRAM de  $K$  palabras de  $n$  bits cada una. Cada palabra está en una dirección a la que se accede directamente a través de un decodificador de  $k$  a  $K$

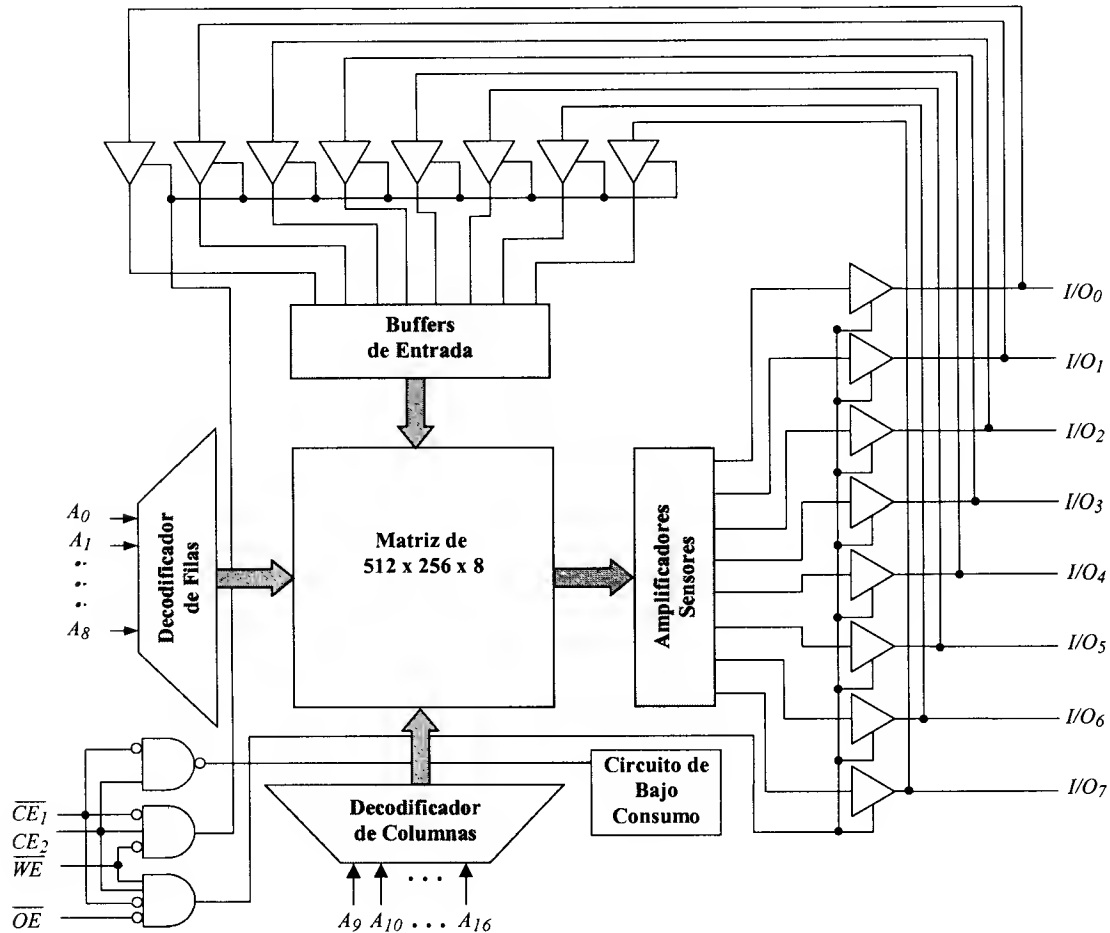
Para ilustrar la descripción funcional de la organización de las memorias SRAM vamos a usar el circuito CY7C109 de CYPRESS. Este circuito es una memoria CMOS estática organizada en 512x256x8. Es decir en 512 filas y 256 columnas y por consiguiente 131072 palabras de 8 bits cuyo diagrama de bloque se muestra en la *figura 11.4*.



**Figura 11.3.** Organización de una memoria SRAM.

Como tiene 512 filas necesitamos 9 bits para direccionarlas ( $A_0, A_1, \dots, A_8$ ). Análogamente, para direccionar las 256 columnas necesitamos los otros 8 bits ( $A_9, A_{10}, \dots, A_{16}$ ). Como las palabras son de 8 bits necesitamos 8 líneas de entrada/salida ( $I/O_0, I/O_1, \dots, I/O_7$ ). La gestión de la función concreta que realizan estas líneas en los distintos modos de funcionamiento se controla a través de

buffers de tres estados, de forma que cuando se facilita la salida de datos ( $\overline{OE}$ ) los buffers de salida están activos y los de entrada están desconectados del bus, en estado de alta impedancia. Inversamente, en operaciones de escritura los datos entran por las mismas líneas físicas pero ahora los buffers de salida están inhibidos (alta impedancia) y los de entrada están facilitados.



**Figura 11.4.** Diagrama de bloques del circuito de memoria SRAM CY7C109 de CYPRESS.

La memoria posee un circuito de control con cuatro señales ( $\overline{CE}_1$ ,  $\overline{CE}_2$ ,  $\overline{WE}$  y  $\overline{OE}$ ) que le permiten operar en cuatro modos: *lectura*, *escritura*, *bajo consumo cuando no está seleccionada* y *seleccionada con las salidas inhibidas*. La tabla de verdad correspondiente a las distintas

configuraciones en los valores de estas variables necesarias para la selección de cada uno de estos modos es la que se muestra en la *figura 11.5*.

$\overline{CE}_1$	$CE_2$	$\overline{OE}$	$\overline{WE}$	$I/O_0 - I/O_7$	<i>Modo</i>	<i>Consumo</i>
H	x	x	x	<i>Alta Z</i>	<i>Bajo Consumo</i>	<i>Standby (<math>I_{SB}</math>)</i>
x	L	x	x	<i>Alta Z</i>	<i>Bajo Consumo</i>	<i>Standby (<math>I_{SB}</math>)</i>
L	H	L	H	<i>Salida de Dato</i>	<i>Lectura</i>	<i>Activa (<math>I_{CC}</math>)</i>
L	H	x	L	<i>Entrada de Dato</i>	<i>Escritura</i>	<i>Activa (<math>I_{CC}</math>)</i>
L	H	H	H	<i>Alta Z</i>	<i>Seleccionada pero con salidas inhibidas</i>	<i>Activa (<math>I_{CC}</math>)</i>

$\overline{CE}_1$  = facilitación del circuito (*chip enable*), activa en baja  
 $CE_2$  = facilitación del circuito (*chip enable*), activa en alta  
 $\overline{OE}$  = facilitación de la salida (*output enable*), activa en baja  
 $\overline{WE}$  = facilitación de la escritura (*write enable*), activa en baja

**Figura 11.5.** Configuraciones de las señales de control para la selección de los distintos modos de funcionamiento de la SRAM.

Para *escribir* una palabra en la memoria seleccionamos la configuración ( $\overline{CE}_1 = L$ ,  $CE_2 = H$ ,  $\overline{OE} = x$  y  $\overline{WE} = L$ ). Es decir, ponemos en baja la señal de selección del circuito en el que vamos a escribir ( $\overline{CE}_1 = 0$  y  $CE_2 = 1$ ) y activamos la orden de escritura ( $\overline{WE} = 0$ ). Entonces los datos que están en las 8 líneas bidireccionales de entrada/salida ( $I/O_0$ , ...,  $I/O_7$ ) se escriben en los 8 biestables que están en la dirección especificada por ( $A_0$ , ...,  $A_{16}$ ).

Para *leer* usamos la configuración de control  $\overline{CE}_1 = L$ ,  $CE_2 = H$ ,  $\overline{OE} = L$  y  $\overline{WE} = H$ , es decir, activamos las líneas de selección del circuito en el que vamos a leer ( $\overline{CE}_1 = 0$  y  $CE_2 = 1$ ), facilitamos la salida de datos ( $\overline{OE} = 0$ ), y damos la orden de lectura ( $\overline{WE} = 1$ ).

Este circuito tiene la posibilidad de reducir drásticamente el consumo cuando no está seleccionado (modo "power down"), pasando de  $770mW$  a  $165mW$ . En este caso la configuración debe ser  $\overline{CE}_1 = 1$ ,  $CE_2 = x$ , ó  $\overline{CE}_1 = x$ ,  $CE_2 = 0$ , no importando los valores de  $\overline{OE}$  y  $\overline{WE}$ . Esto

hace que las líneas de entrada/salida pasen al estado de alta impedancia.

Finalmente, el circuito puede estar seleccionado ( $\overline{CE_1} = 0$ ,  $CE_2 = 1$ ) pero no tener facilitadas las salidas ( $\overline{OE} = 1$  y  $\overline{WE} = 1$ ) y consumiendo como en los modos de lectura y escritura.

La *figura 11.6* muestra los diagramas de tiempo correspondientes a los ciclos de lectura y escritura<sup>(\*)</sup>. Los parámetros que miden la duración de un ciclo de lectura o escritura son  $t_{RC}$  (read cycle time) y  $t_{WC}$  (write cycle time) y los circuitos se caracterizan por el valor mínimo de estos tiempos. En el circuito usado como ejemplo es  $t_{RC} = t_{WC} = 20ns$ . Todo ciclo de lectura o escritura comienza estableciendo una dirección válida en la que leer o escribir y esta dirección debe permanecer estable durante todo el intervalo de tiempo necesario para que el resto de las señales de control operen. El parámetro asociado es  $t_{AA}$  (address to data valid) y su valor máximo (20ns en el ejemplo) coincide con el valor mínimo del tiempo de ciclo,  $t_{RC}$ .

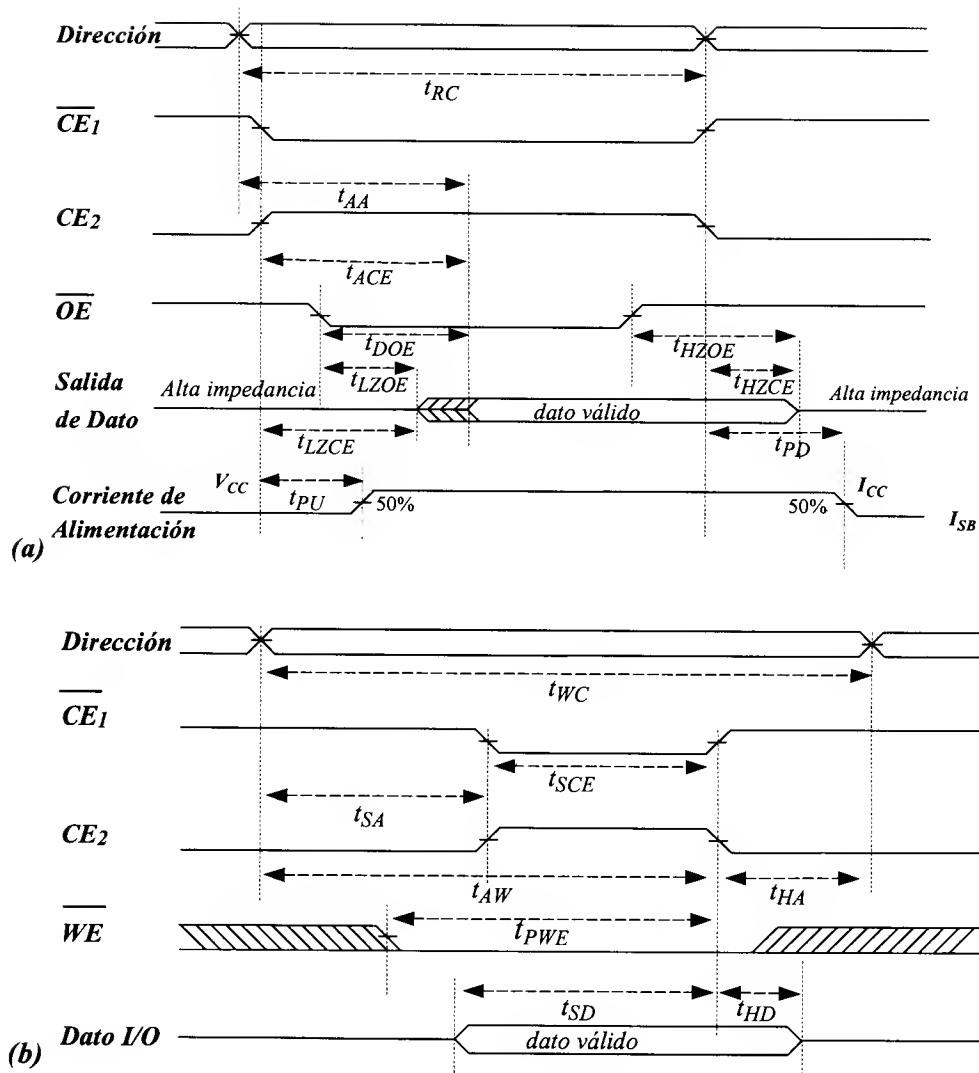
Tras el inicio del intervalo  $t_{RC}$  y  $t_{AA}$ , cuando la dirección está estable, actúan las señales de selección del circuito ( $\overline{CE_1}$  pasa a baja, donde es activa y  $CE_2$  pasa a alta). A partir de aquí las cosas son diferentes en lectura y en escritura.

En lectura, tras saber en qué circuito y en qué dirección de ese circuito tenemos que leer, se activa la facilitación de las líneas de salida ( $\overline{OE}$  pasa a baja, donde es activa) y un cierto tiempo después tenemos en el bus los datos correspondientes. Este intervalo de tiempo se mide con tres parámetros:  $t_{ACE}$ ,  $t_{DOE}$  y  $t_{LZOE}$ <sup>(\*\*)</sup>. El primero,  $t_{ACE}$  es el intervalo temporal que transcurre desde que  $\overline{CE_1}$  pasa a baja (y  $CE_2$  a alta) hasta que aparecen en el bus de salida los datos leídos y estos son estables, pudiendo por consiguiente considerarse como válidos. Análogamente  $t_{DOE}$  es el intervalo temporal desde que se facilita la salida ( $\overline{OE}$  en baja) hasta que hay datos de salida válidos. Finalmente,  $t_{LZOE}$  mide el tiempo entre la bajada de  $\overline{OE}$  y el final del estado de alta impedancia en los buffers de tres estados donde van a aparecer los datos..

---

<sup>(\*)</sup> Conviene recordar ahora el criterio sobre la representación de cronogramas que presentamos en el tema 7, al hablar de los ciclos de lectura y escritura en memorias EEPROM.

<sup>(\*\*)</sup> Obsérvese que al igual que en las EEPROM hemos mantenido la nomenclatura inglesa para los subíndices de los parámetros temporales porque es descriptiva de su significado.



**Figura.11.6,** Cronogramas de los ciclos de lectura (a) y escritura (b) de una memoria SRAM.

Si nos fijamos ahora en el cronograma correspondiente a un ciclo de escritura (parte (b) de la figura 11.6), vemos tras la obtención de una dirección válida y el establecimiento de la orden de escritura ( $\overline{WE}$  en baja) aparecen los datos válidos (estables) y las órdenes de selección de circuito ( $\overline{CE}_1$  en baja y  $CE_2$  en alta). Los intervalos temporales de interés ahora son:



$t_{SCE}$  = Duración de la facilitación, desde la bajada de  $\overline{CE_I}$ , hasta su subida.

$t_{SA}$  = Intervalo desde que la dirección está estable hasta que comienza la escritura.

$t_{AW}$  = Intervalo desde que la dirección está estable hasta que finaliza la escritura.

$t_{SD}$  = Intervalo desde el establecimiento de los datos hasta el final de la escritura.

$t_{HD}$  = Tiempo que se retienen los datos tras el final del proceso de escritura. Después, las líneas de entrada/salida pasan al estado de alta impedancia.

No importa mucho el circuito concreto que hemos seleccionado para comentar la organización de las memorias SRAM. Lo importante es comprenderlo, porque con ligeras modificaciones será válido para otros muchos circuitos SRAM.

### 11.3. EVOLUCIÓN DE LAS SRAM

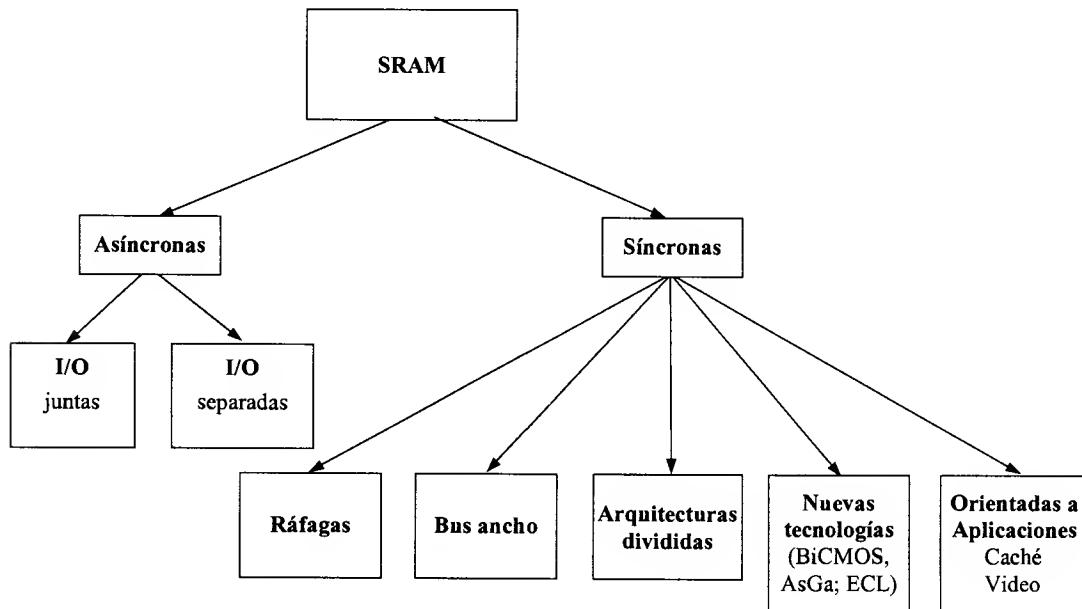
Hablar de evolución en las SRAM es hablar de las distintas estrategias seguidas para aumentar su velocidad y tamaño siendo todavía competitivas en precio con las RAM dinámicas. Cada vez se piden más prestaciones de las memorias RAM estáticas en todos sus dominios de aplicación: como memoria caché, en el diseño de subsistemas gráficos y en equipos de comunicación de alta velocidad, donde todavía hay diferencias entre la velocidad a la que el procesador necesita acceder a los datos y la velocidad intrínseca a la tecnología y arquitectura de los circuitos SRAM que sirven a ese procesador.

Para conseguir aumentar la velocidad de las SRAM se han introducido las siguientes modificaciones sobre las versiones asíncronas básicas:

1. Hacer que la memoria sea síncrona (quedando así bajo el control del reloj general del sistema) y con posibilidad de acceso por “ráfagas” a varias direcciones de memoria (generadas internamente de forma automática) a partir de un acceso inicial convencional a una determinada dirección de inicio de la ráfaga.
2. Actuar sobre la arquitectura de la memoria, usando líneas separadas para entrada y salida, aumentando el ancho de banda del bus interno y añadiendo señales externas de control (*output enable*) que permiten tener un circuito direccionado y seleccionado antes de activar las entradas.

3. Usar arquitecturas divididas en varias zonas para que el camino que es necesario recorrer para alcanzar una celda concreta sea, en valor medio, más corto.
4. Desarrollar nuevas tecnologías, en adición o alternativa a la CMOS actual. Por ejemplo, BiCMOS, Arseniuro de Galio (AsGa) o bipolar ECL. Estas tecnologías son más rápidas, ofrecen mayores niveles de corriente pero también consumen más y son más caras.
5. Desarrollar memorias SRAM orientadas a aplicaciones específicas, por ejemplo para su uso como memoria caché de correspondencia directa con la memoria principal o con distinto grado de asociatividad, o para aplicaciones de video.

El esquema de la *figura 11.7* resume estas extensiones de las memorias SRAM

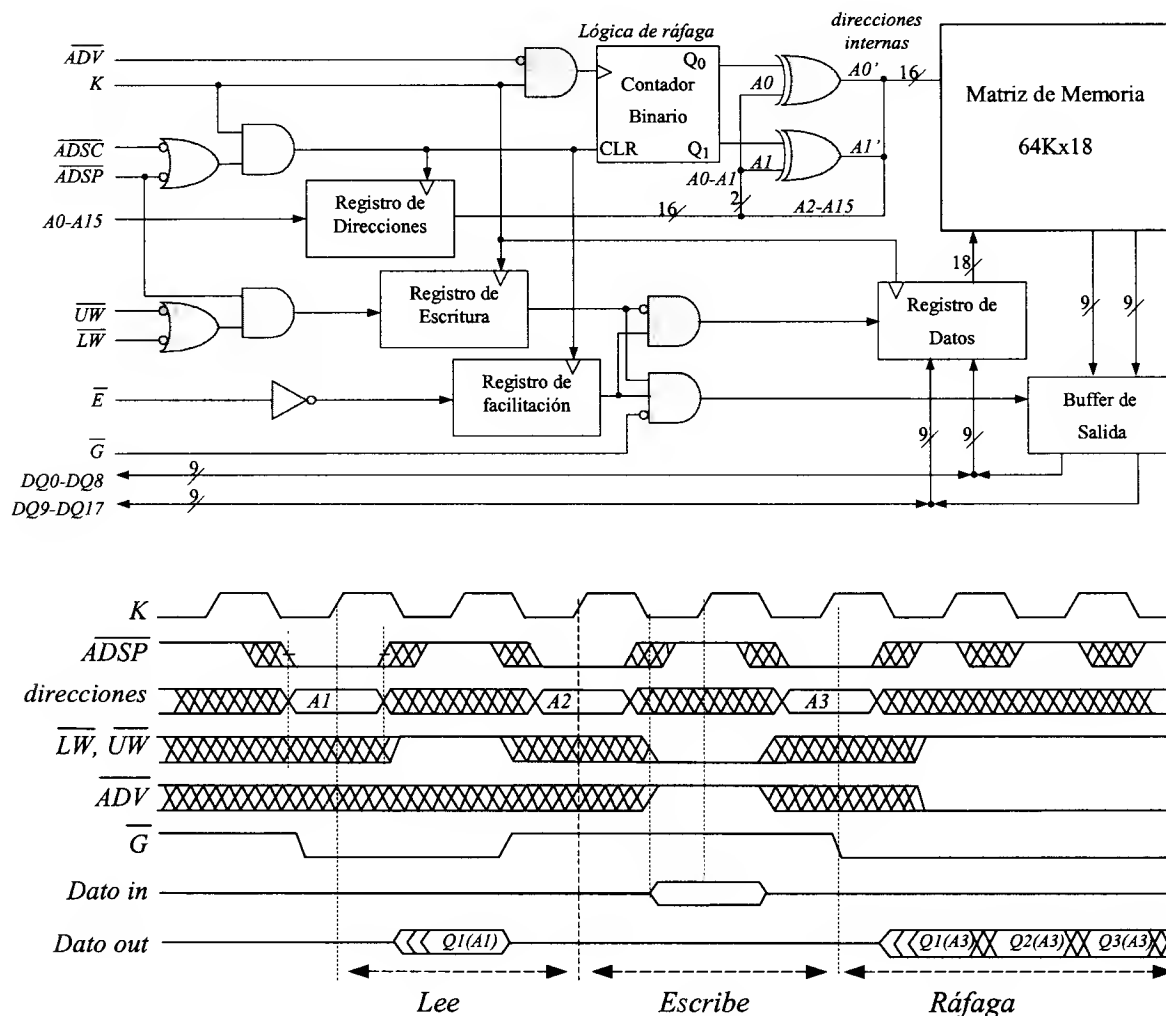


**Figura 11.7.** Evolución de las memorias SRAM para aumentar la velocidad intrínseca y la del sistema que la incorpora (arquitectura).

De todas estas innovaciones vamos a exponer con cierto detalle sólo las SRAM síncronas de ráfaga que incluyen dos mecanismos básicos:

- a) El uso de registros y el comportamiento síncrono.

- b) El acceso a las direcciones por ráfagas ("burst"). El resto creemos que queda fuera del alcance de este texto. La figura 11.8 muestra un esquema simplificado de una SRAM síncrona de 32Kx8.



**Figura 11.8.** Esquema simplificado de una hipotética SRAM síncrona y con posibilidad de acceso a las direcciones en modo ráfaga.

La diferencia esencial entre una memoria síncrona y una asíncrona es la que ya conocemos con carácter general para todos los circuitos secuenciales. En una SRAM asíncrona los procesos de

lectura/escritura ocurrían tras un cierto tiempo de asentamiento de los niveles de tensión en las distintas señales de control. Ahora, al hacer síncrona a la SRAM todo tiene que ocurrir en los flancos de subida o bajada de los pulsos del reloj general del sistema en el que está integrada esa SRAM. Por ejemplo, en un sistema de cálculo. Es decir, la SRAM queda bajo el control del reloj general del procesador, disminuyendo o eliminando los intervalos temporales de espera.

Para hacer síncrona la SRAM necesitamos introducir *registros* en las entradas de *direcciones*, *datos* y *señales de control* (selección de circuito y orden de lectura/escritura) y en los datos de salida. Estos registros permiten la sincronización de la memoria porque en los flancos positivos de los pulsos de reloj se captan todos los valores y se almacenan en los biestables del registro, sin necesidad de que permanezcan retenidos más tiempo en espera del asentamiento. En el siguiente pulso de reloj (flanco positivo), los contenidos de esos registros pasan a operar sobre el resto del circuito SRAM, que es prácticamente igual al del caso asíncrono. Es decir, incluye los decodificadores de direcciones, el núcleo de celdas RAM estáticas, los amplificadores sensores y la acción de control de las señales de selección de circuito ( $\overline{CS}$ ) y de activación de los procesos de lectura/escritura ( $\overline{WE}$ ). Y lo mismo ocurre con la forma de operar a partir de los registros. En un ciclo de lectura se capta primero la dirección y se almacena en el registro durante el flanco positivo de un pulso de reloj y después se decodifica; a continuación aparecen los contenidos en el registro de datos de salida. En el siguiente pulso, los datos del registro pasan a los buffers. En algunos casos las llamadas SRAM síncronas no lo son totalmente porque tienen un terminal (*output enable*) que es asíncrono para proporcionar facilidades adicionales de control sobre las salidas.

En algunas SRAM síncronas se aumenta la velocidad haciendo que el bus de datos opere al doble de la velocidad del reloj, sincronizando con ambos flancos del pulso (subidas y bajadas) y usando un bus interno más ancho que acepta la obtención de dos palabras en cada pulso de reloj.

El segundo mecanismo que queremos comentar es el *acceso a las direcciones por ráfagas*. Este modo de acceso lo incorporan en general tanto las SRAM como las RAM dinámicas, siempre que sean síncronas y es compatible con la mayoría de los procesadores. Es una forma muy rápida de acceder de una vez a la información almacenada en un conjunto pequeño de direcciones próximas. Para ello, se accede a la primera de estas direcciones por el procedimiento normal y después se accede al resto de las direcciones de la ráfaga a través de un mecanismo interno que las genera usando un contador de dos bits que, a partir de los dos bits menos significativos del registro de direcciones,  $A0$ ,  $A1$ , genera la secuencia 00,01,10,11. Al ser generadas internamente, el acceso a estas direcciones es

mucho más rápido.

Por ejemplo, en una SRAM de ráfaga CY7C178, la secuencia de direcciones generadas por el contador a partir de cada una de las cuatro posibles direcciones iniciales es:

Primera Dirección	Segunda Dirección	Tercera Dirección	Cuarta Dirección
$A_{x+1}, A_x$	$A_{x+1}, A_x$	$A_{x+1}, A_x$	$A_{x+1}, A_x$
00	01	10	11
01	00	11	10
10	11	00	01
11	10	01	00

de forma que todos los datos situados en esta serie de direcciones se pueden empezar a buscar desde la llegada de la primera dirección.

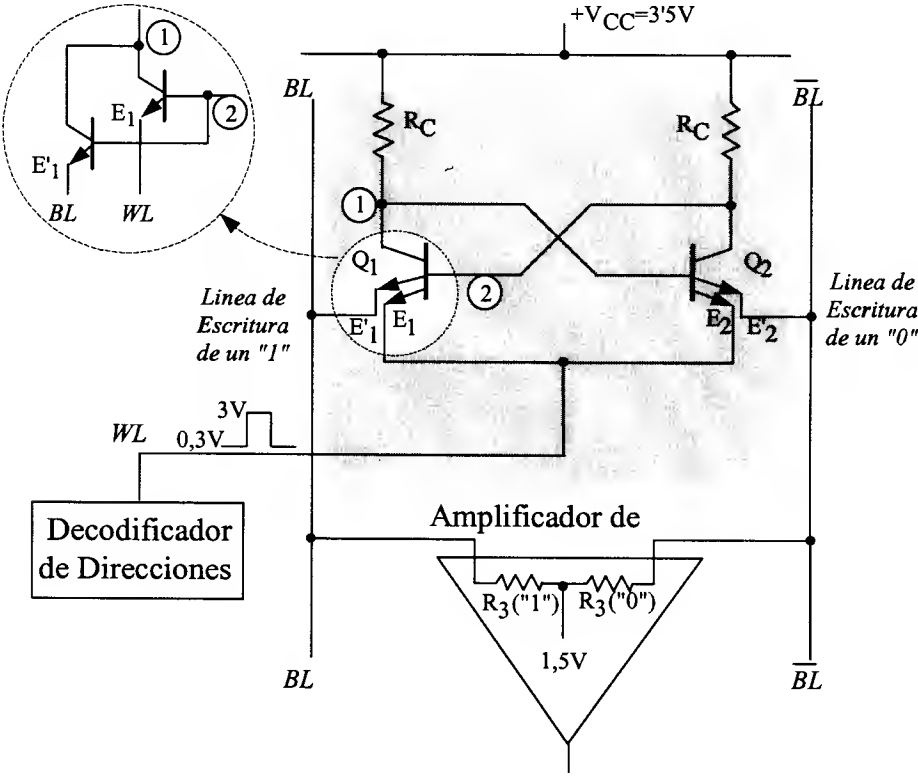
## 11.4. CELDAS RAM ESTÁTICAS (SRAM) EN TECNOLOGÍA BIPOLAR

Ya hemos dicho que las celdas RAM son biestables simplificados al máximo para aumentar la densidad de integración con las facilidades adicionales para permitir la organización expuesta en el apartado anterior, es decir el direccionamiento, la lectura y la escritura. La función que realizaba el pulso de reloj en un biestable R-S sincronizado a nivel la realiza aquí la línea de selección de bit que facilita la entrada y/o salida de datos sólo en la celda direccionada. Estas celdas se pueden realizar en tecnologías bipolar, MOS, CMOS y BiCMOS. Cuando la velocidad es el parámetro dominante se elige bipolar o BiCMOS y cuando interesa aumentar la capacidad se elige CMOS.

La *figura 11.9*, muestra el esquema de una celda bipolar realizada mediante dos transistores multiemisor. Sobre la estructura mínima de dos inversores realimentados (biestables), se superponen dos nuevos emisores para conectar las líneas de bit y se utiliza el circuito de polarización para seleccionar la celda en la que se quiere leer o escribir. Para ello, los emisores  $E_1$  y  $E_2$  se apoyan sobre el pulso de selección de celda, lo que hace que su valor cambie de 0,3 V a 3 V.

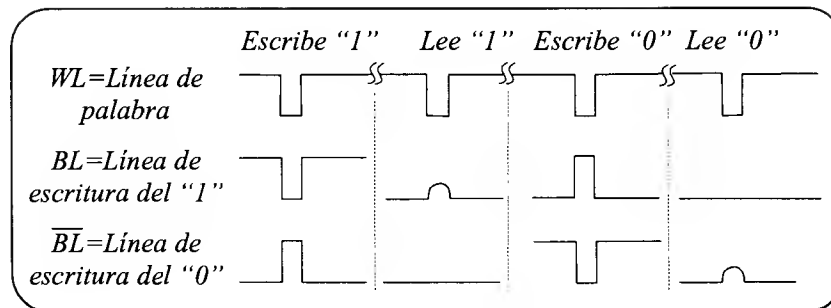
Elegimos el convenio de decir que la celda almacena un “1” cuando el transistor  $Q_1$  está conduciendo ( $Q_2$  al corte) y un “0” cuando  $Q_2$  está conduciendo ( $Q_1$  al corte). Por consiguiente, el emisor adicional de  $Q_1$  ( $E_1'$ ) está conectado a la línea de escritura de un “1” y el emisor adicional de  $Q_2$  ( $E_2'$ ), a la línea de escritura de un cero. Estas líneas van a otras muchas celdas pero sólo operan

sobre la celda que está seleccionada. Finalmente, el resultado de la lectura aparece sobre un amplificador diferencial que conecta las líneas de lectura de “1” y lectura de “0”. En su mínima expresión, este sensor del estado se limita a dos resistencias,  $R_3$ , conectadas entre las líneas de lectura y una tensión fija de 1,5.voltios. La *figura 11.10* muestra el cronograma de la línea de palabra y de las líneas de bit con los valores correspondientes en los intervalos de lectura y escritura de “1” y “0”.



<i>Escribe "1"</i>	<i>Escribe "0"</i>	<i>Lee "1"</i>	<i>Lee "0"</i>	<i>Seleccionada</i>	<i>No Seleccionada</i>	<i>Alamacena "1"</i>	<i>Alamacena "0"</i>
$BL=0V$ $\overline{BL}=3,5V$	$BL=3,5V$ $\overline{BL}=0V$	$V[R_3("1")]>0$ $V[R_3("0")]\approx 0$	$V[R_3("1")]>0$ $V[R_3("0")]\approx 0$	$WL=3V$	$WL=0,3V$	$Q_1 \rightarrow ON$ $Q_2 \rightarrow OFF$	$Q_1 \rightarrow OFF$ $Q_2 \rightarrow ON$

**Figura 11.9.** Circuito básico representativo de las celdas RAM estáticas realizadas con transistores bipolares multiemisor.



**Figura 11.10.** Cronograma correspondiente a la celda RAM estática de la figura anterior.

Para *seleccionar* la celda el decodificador de direcciones manda un impulso positivo de 3V por la línea  $WL$  que en estado de reposo tenía un valor bajo, del orden de 0,3V. En estas condiciones el transistor  $Q_1$  continúa en conducción ( $Q_2$  en corte) pero ahora su emisor activo es  $E'_1$  (a un potencial de 0,5V). Por consiguiente, la información se conserva.

Para *leer*, se selecciona la celda de la forma descrita anteriormente. Sólo el emisor del transistor que conduce ( $Q_1$  en el ejemplo) proporciona corriente a una de las líneas de datos produciendo un pulso a la salida del amplificador de lectura. En el caso mínimo de la figura, este pulso aparece sobre la resistencia  $R_3$ ("1") del lado correspondiente. Si hubiera estado activo  $Q_2$  (cuando la celda almacena un "0"), este pulso habría aparecido sobre la línea  $\overline{BL}$  y sobre la resistencia  $R_3$ ("0").

Para *escribir* un dato en la celda se comienza direccionándola de la misma forma que en el proceso de lectura, y se introduce el dato que queremos almacenar por la línea de bit correspondiente. Supongamos que queremos escribir la información complementaria a la almacenada por la celda. Para ello, mandamos un pulso por  $WL$  y simultáneamente bajamos la tensión en la línea  $\overline{BL}$  manteniendo en alta la de  $BL$ . Se produce así una diferencia de tensión entre ambas líneas haciendo pasar  $Q_2$  a conducción y por procesos regenerativos arrastrando a  $Q_1$  al corte, con lo que se ha hecho volcar al biestable.

Las celdas SRAM en tecnología bipolar tienen su campo natural de aplicación en el diseño de las memorias caché de computadores de altas prestaciones en los que la velocidad es el parámetro esencial (tiempos de ciclo inferiores a 5ns). Sin embargo presentan problemas inherentes a la tecnología. En particular, la baja impedancia de los transistores bipolares en comparación con los valores usuales en MOS y CMOS. Esto produce dos efectos. Por un lado, hace muy difícil la

obtención de celdas bipolares dinámicas (almacenamiento en un condensador aislado), obligando al uso de la realimentación (circuitos R-S disparados a nivel) y por otro lado, la baja calidad de la función de conmutación necesaria en el proceso de selección de una celda concreta para lectura/escritura y en el aislamiento de todas las demás.

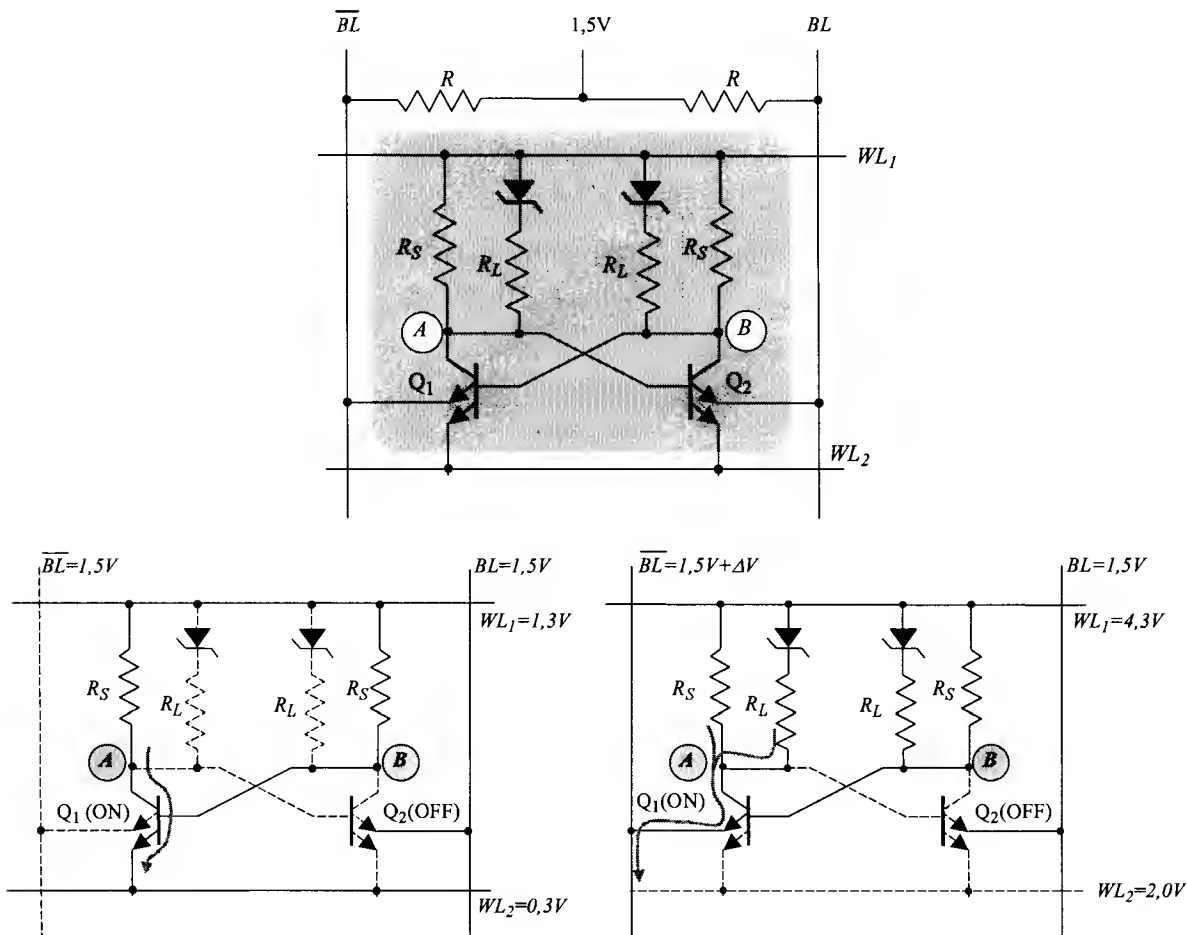
Para resolver en parte este problema de aislamiento limitado debido a la baja impedancia relativa se usan diodos de barrera Schottky, (SBD, Schottky barrier-diode) y líneas dobles para la selección de bit ( $WL_1$  y  $WL_2$ ), tal como se muestra en los circuitos de la *figura 11.11*. Estos circuitos se llaman SBD-SRAM, para indicar la inclusión de los diodos Schottky.

El resto del circuito de estas celdas SRAM es el convencional que ya conocemos. Es decir están formadas también por dos inversores acoplados en realimentación en los que no sólo se polarizan sobre pulsos los emisores como en el caso anterior ( $WL_2=0,3V$  ó  $WL_2=2V$ ), sino que también se usan pulsos para polarizar los colectores ( $WL_1=1,3V$  ó  $WL_1=4,3V$ ). Las líneas de bit ( $BL$  y  $\overline{BL}$ ) del circuito de la *figura 11.11* están sujetas a una fuente de continua de  $1,5V$ , en la misma configuración que en la celda anterior (*figura 11.10*) y su comportamiento global también es análogo.

Cuando la celda no está seleccionada, parte (b) de la *figura 11.11*, los transistores están polarizados con  $1V$  ya que  $WL_1=1,3V$  y  $WL_2=0,3V$ . Este voltaje es suficiente para mantener el estado de la celda (un “1” en este caso), aunque reduce el consumo. A esta disminución del consumo también contribuye la inhibición de la componente más pequeña de la resistencia de carga,  $R_L$ , con la ayuda del diodo Schottky, dejando sólo la resistencia más grande,  $R_S$ , lo que disminuye el valor de la corriente de colector y, por consiguiente, el consumo. Dado que las líneas de bit ( $BL$  y  $\overline{BL}$ ) están sujetas a  $1,5V$ , los segundos emisores de  $Q_1$  y  $Q_2$  están al corte, con lo que efectivamente la celda está aislada.

En el modo de *lectura*, parte (c) de la *figura 11.11*, las líneas de palabras,  $WL_1$  y  $WL_2$ , pasan a sus valores altos de potencial ( $WL_1=4,3V$  y  $WL_2=2V$ ) y los diodos Schottky pasan a conducción reduciendo la resistencia efectiva de carga ( $R_L$  en paralelo con  $R_S$ , que es mucho menor) y aumentando el valor de la corriente de colector disponible. Los emisores conectados a  $WL_2$  pasan al corte porque las líneas de bit están en el valor bajo de tensión y la corriente de emisor del transistor  $Q_1$ , que es el que conduce en este caso, va hacia la línea de bit  $\overline{BL}$  provocando un “incremento de tensión” en la resistencia que apoya  $\overline{BL}$  con  $1,5V$ . Este cambio de tensión es el que detecta el amplificador sensor.





**Figura 11.11.** Celdas de RAM estática con diodos de barrera Schottky (SBD SRAM). **(a)** Circuito básico. **(b)** Partes activas e inactivas durante el modo de reposo, cuando la celda está aislada y almacenando un "1" (por convenio decimos que almacena un "1" cuando  $Q_1$  conduce y  $Q_2$  no conduce y almacena un "0" cuando  $Q_1$  no conduce y  $Q_2$  conduce). **(c)** Partes activas (*trazo continuo*) e inactivas (*trazo discontinuo*) en modo de lectura, cuando la celda está seleccionada y se está leyendo un "1".

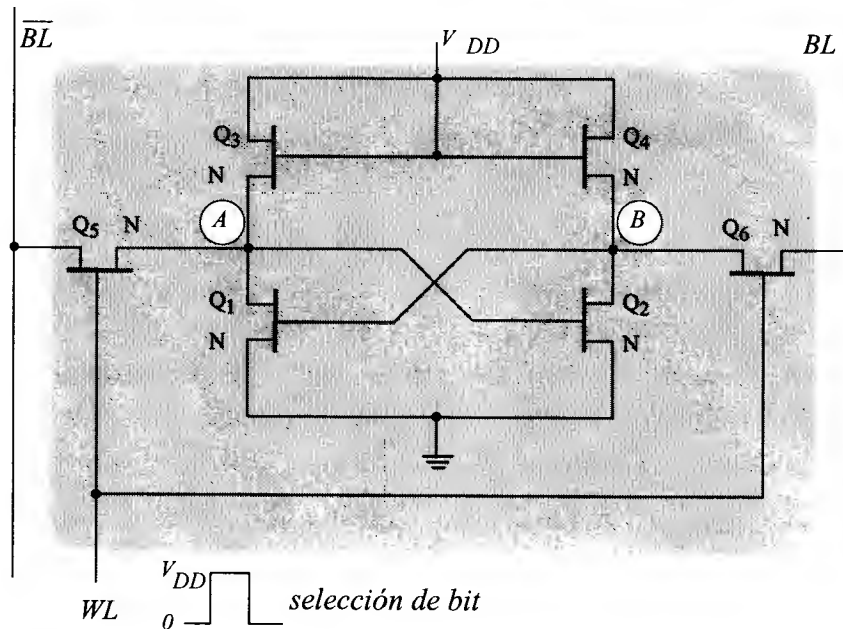
Finalmente, para escribir un cero en la celda SRAM la línea de bit ( $BL$ ) se lleva a 0V estando en alta a la vez las dos líneas de palabras ( $WL_1 = 4.3V$  y  $WL_2 = 2V$ ). En estas condiciones de polarización el transistor  $Q_2$  pasa a conducción y cambia el estado almacenado en la celda (de "1" a

“0”), tras lo cual las líneas de palabras vuelven a su potencial de reposo.

Aunque siguen desarrollándose celdas SRAM, en tecnología bipolar para aumentar la velocidad, su importancia ha disminuido con la llegada de la tecnología BiCMOS que combina la velocidad de la bipolar con la alta densidad de integración de las tecnologías MOS y CMOS.

## 11.5. CELDAS RAM ESTÁTICAS (SRAM) EN TECNOLOGÍA MOS

La idea básica del diseño de celdas MOS estáticas es la misma que en el caso bipolar. Partimos de un biestable MOS con dos inversores acoplados ( $Q_1$ - $Q_3$  y  $Q_2$ - $Q_4$ ) y le añadimos otros dos ( $Q_5$  y  $Q_6$ ) para las operaciones de *selección*, *lectura* y *escritura*. La *figura 11.12* muestra el circuito correspondiente para esta celda de 6 transistores NMOS de realce.



**Figura 11.12.** Celda NMOS para RAM estática.

Supongamos al igual que en el caso bipolar que la celda *almacena un “1”* cuando  $Q_1$  está en *conducción* y  $Q_2$  en *corte* y viceversa en el caso de almacenar un “0”. Cuando la celda no está direccionada está aislada del resto de la memoria porque los transistores  $Q_5$  y  $Q_6$  son puertas de transferencia abiertas (muy alta impedancia) debido a la polarización que reciben a través de la línea

$WL$  procedente del decodificador de direcciones.

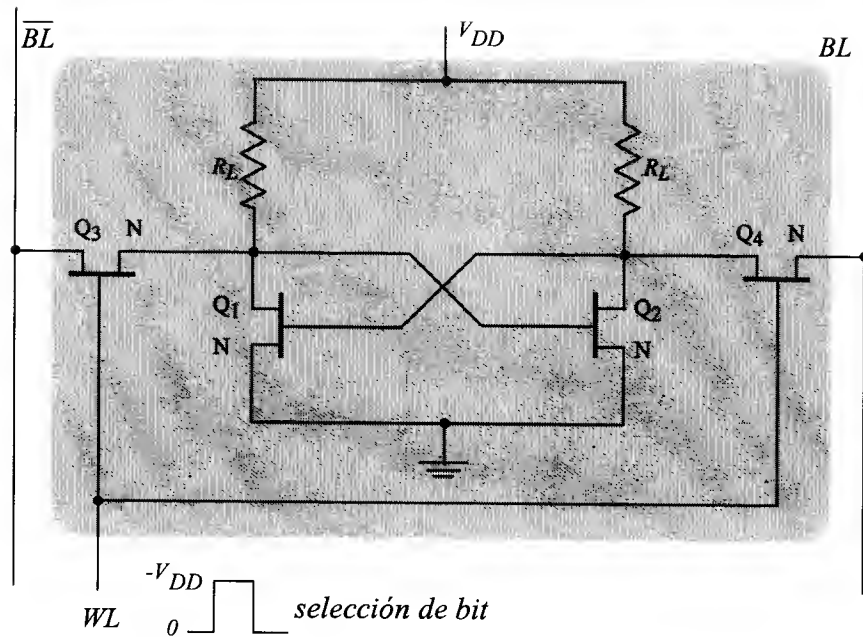
Para *leer* el contenido de la celda se activan sus puertas de transmisión poniendo  $V_{DD} = 12V$  en  $WL$ , con lo que  $Q_5$  y  $Q_6$  conducen y transmiten el potencial del punto A (punto de “0”) a la línea de datos correspondiente ( $\overline{BL}$ ).

Análogamente, el potencial del punto B (punto de “1”), se transmite a  $BL$  y el amplificador diferencial de lectura extrae el dato. Obsérvese que la celda es totalmente simétrica y que sólo el convenio inicial (“1” significa  $Q_1$  en conducción) y el etiquetado de las líneas de datos ( $BL$  es la línea de “1”), permiten la distinción. Cuando la celda contenía un “1”, al final de la operación de lectura aparecen  $12V$  en  $BL$  y  $0V$  en  $\overline{BL}$ . Cuando tenía un “0”, aparece  $0V$  en  $BL$  y  $12V$  en  $\overline{BL}$ .

Para *escribir* en la celda se comienza direccionándola mediante un pulso positivo en  $WL$ , como en el caso anterior. Esto abre de nuevo las puertas de transmisión  $Q_5$  y  $Q_6$  y permite conmutar el biestable. Para *escribir un “1”* se mantiene  $BL$  en  $12V$  y se baja la tensión de  $\overline{BL}$  a  $0V$ . Esto hace conducir a  $Q_1$  a través de  $Q_6$  (si es que estaba cortado), quedando el drenador de  $Q_1$  (punto A de la *figura 11.12*) a  $0V$ . Como esta es la tensión del terminal de puerta de  $Q_2$ , este pasa al corte. Cuando  $WL$  vuelve a  $0V$ , el “1” lógico queda almacenado en la celda dado que los transistores  $Q_5$  y  $Q_6$  pasan de nuevo al estado de corte presentando alta impedancia y dejando a la celda aislada.

La celda anterior necesita 6 transistores. Una versión con sólo cuatro transistores es la que se muestra en la *figura 11.13*. Esta celda SRAM de carga resistiva procede de la simplificación de las celdas NMOS donde se han sustituido el par de inversores NMOS con carga activa y acoplados, por un par de inversores NMOS con carga resistiva. Así, al cambiar los transistores NMOS por resistencias  $R_L$ , el tamaño de la celda se reduce.

El parámetro crítico en el diseño de estas celdas es el valor de la resistencia que debe de ser “suficientemente alto” para conseguir un buen margen de ruido y un bajo consumo y “suficientemente bajo” para no deteriorar el valor de los retardos de propagación en las transiciones de baja a alta. Este segundo punto se soluciona parcialmente precargando las líneas de bit a  $V_{DD}$  de forma que las transiciones baja a alta en las líneas de bit sólo ocurren durante la precarga y no durante la operación de lectura. Así, las cargas resistivas  $R_L$  nunca tienen que suministrar corriente durante una transición. Los detalles tecnológicos que hacen especialmente útiles a estas celdas en memorias SRAM quedan fuera del alcance de este texto.

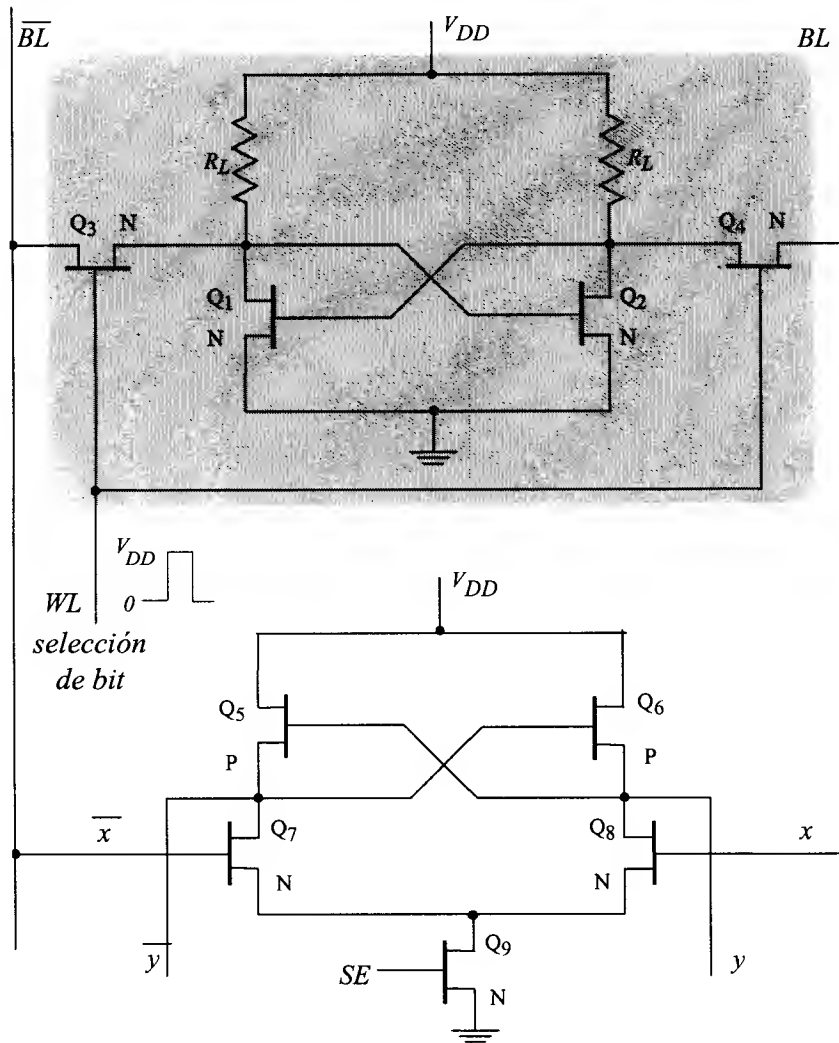


**Figura 11.13.** Celda SRAM de cuatro transistores con carga resistiva ( $R_L$ )

Hasta ahora, los esfuerzos realizados en el diseño de las memorias RAM estáticas estaba encaminado a aumentar la densidad de integración del núcleo de celdas biestables donde se almacenan los datos. Sin embargo, para direccionar, leer y escribir esos datos nos hace falta un conjunto de circuitos periféricos adicionales tales como *decodificadores*, *amplificadores sensores* y *circuitos de temporización y control* de los que depende en gran medida la velocidad y eficacia global de un circuito de memoria. Todos estos circuitos han sido estudiados en los capítulos 5, 6, 7 y 10, salvo los *amplificadores sensores* que vamos a comentar aquí.

La importancia de los amplificadores sensores se debe a lo reducido del salto de tensión medido en la operación de lectura, a la necesidad de ofrecer niveles de corriente superiores a los del interior de la celda para acelerar las transiciones en las líneas de bit y al problema del ruido que se superpone a los niveles de señal. Afortunadamente, las celdas SRAM admiten amplificadores diferenciales como sensores porque tienen salida diferencial ( $BL$  y  $\overline{BL}$ ) con lo que el factor de rechazo al modo común es alto y sólo se amplifican las diferencias entre las dos señales de entrada. Además, un mismo amplificador diferencial puede compartirse entre varias columnas insertando los transistores de paso del decodificador de columnas entre las celdas de memoria y el diferencial.

La *figura 11.14* muestra el circuito de un posible sensor diferencial para celdas SRAM. Las líneas de bit se conectan a las entradas  $x$  y  $\bar{x}$  del amplificador diferencial que usa como cargas activas otros dos transistores PMOS ( $Q_5$  y  $Q_6$ ) acoplados en realimentación positiva. Hay otras muchas configuraciones diferenciales usando transistores PMOS y NMOS que son también válidas.



**Figura 11.14.** Uso de amplificadores diferenciales como sensores en celdas SRAM.

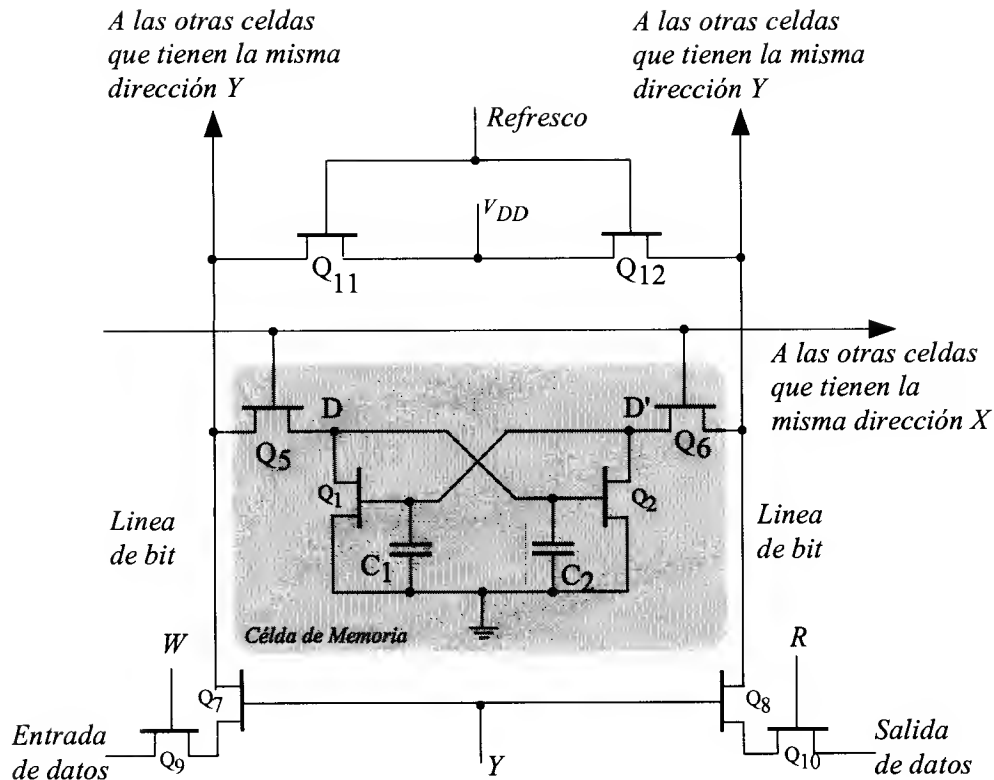
## 11.6. CELDAS RAM DINÁMICAS (DRAM) EN TECNOLOGÍA MOS

Cuando se desea alta capacidad y bajo consumo con velocidades algo más moderadas que las de RAM estática, se elige como solución las memorias RAM dinámicas. Independientemente de las variaciones que se introducen a nivel de organización y de direccionamiento, aquí nos interesa hablar sólo de la celda básica. La idea esencial es reducir al mínimo el área de integración y el consumo de una celda de memoria RAM pasando de 6 a 4 transistores, después a 3 y finalmente a sólo un transistor. La tecnología MOS es especialmente adecuada para realizar esta minimización por su alta impedancia y su capacidad para inyectar, confinar, mantener temporalmente y extraer paquetes de carga en estructuras capacitivas superficiales. En realidad basta con poder controlar y detectar la presencia de un paquete mínimo de carga en un condensador para interpretar que ahí se almacena un cero o un uno. Sin embargo, a pesar de las barreras activas que se pueden generar en una estructura MOS siempre hay pérdidas de carga de forma que al cabo de un cierto tiempo el paquete de carga desaparece y se borra la información almacenada. Para resolver este problema se introducen circuitos de “*refresco*” que reconstruyen la información de forma periódica, antes de que se pierda. El diseño LSI actual ofrece soluciones monolíticas al problema del refresco, como veremos más adelante.

Ahora nos interesa seguir el camino histórico mostrando distintas soluciones a las celdas RAM dinámicas. La *figura 11.15.a* muestra una celda MOS para RAM dinámica con cuatro transistores.

Si se compara el circuito de la *figura 11.15* con el correspondiente para RAM estática se observa que se han eliminado dos transistores y ahora el estado de la celda se almacena en las capacidades de puerta ( $C_1$  y  $C_2$ ) de los transistores  $Q_1$  y  $Q_2$ . Para evitar una pérdida excesiva de carga en estas capacidades se introduce un mecanismo de refresco que periódicamente lee el estado y lo vuelve a escribir, recuperándose así toda la carga perdida.

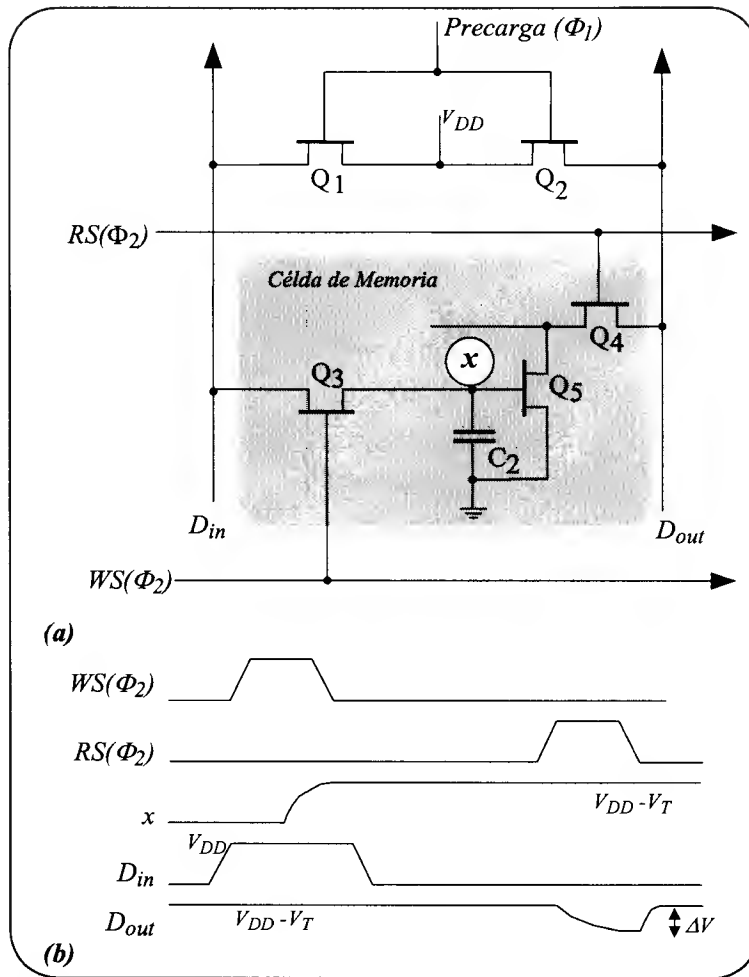
La celda de cuatro transistores se obtuvo a partir de la de 6 al darse cuenta los diseñadores, primero, que las resistencias de carga podían sustituir a dos transistores y, después, que la única función de estas resistencias de carga era “rellenar” la celda con la carga perdida a través de las corrientes de fugas. Por eso podían eliminarse esas resistencias y, “simplemente”, reponer de cuando en cuando la carga perdida mediante la reescritura periódica del contenido de la celda. Así, en general, la reducción en complejidad y área de semiconductor conseguida compensaba la complejidad y área adicionales necesarias ahora para implementar los mecanismos de refresco.



**Figura 11.15.** Celdas RAM dinámicas. (DRAM) de 4 transistores.

Un paso posterior fue pasar de cuatro a tres transistores. Esta simplificación nace al observar que, de hecho, la celda de cuatro transistores es redundante porque almacena a la vez el valor del dato y su complementario, como en todos los biestables. Al eliminar un transistor, se elimina esta redundancia y se reduce el área de integración. Se obtiene así un circuito como el que se muestra en la *figura 11.16*, en el que se han separado los caminos de lectura y escritura y se usa un mecanismo de precarga. La información sigue almacenándose a través del valor de la carga almacenada en el condensador  $C_2$  y del transistor de carga  $Q_5$  que está en conducción o en corte dependiendo del valor de carga en  $C_2$ , ya que la tensión en  $C_2$  es su tensión entre fuente y puerta.

La línea de bit ( $D_{out}$ ) está precargada a  $V_{DD}$  ó a  $V_{DD}-V_T$ , dependiendo del estado del transistor de precarga  $Q_2$ .



**Figura 11.16.** Celda DRAM con tres transistores MOS. (a) circuito. (b) Formas de onda durante los ciclos de lectura y escritura.

El transistor de carga,  $Q_5$ , está en serie con  $Q_4$  y este se apoya en  $D_{out}$  de forma que dependiendo del valor de la carga en  $C_2$  (dato almacenado), y por consiguiente, del estado de  $Q_5$ , la combinación serie de  $Q_5$  y  $Q_4$  hace bajar el potencial de  $D_{out}$  o lo deja como está, dando lugar a un proceso de lectura. Esta celda opera en ciclos de dos fases. Durante la primera mitad de cada ciclo de lectura o escritura se realiza una precarga de las líneas  $D_{in}$  y  $D_{out}$  a través de  $Q_1$  y  $Q_2$ . Al ser la línea  $D_{in}$  altamente capacitiva la transferencia de un “1” a  $C_1$  no reduce el nivel de precarga de forma significativa. Para escribir un “0” se descarga el condensador en la línea. El ciclo de lectura se realiza



pasando  $Q_4$  a conducción al final de la precarga. Obsérvese que esta celda es inversora, es decir el valor que se sensa en la línea  $D_{out}$  es el inverso del almacenado en  $C_2$ . El mecanismo de refresco usado para recuperar la carga perdida por corrientes de fugas consiste en leer el dato almacenado, poner su inverso en la línea de entrada de datos ( $D_{in}$ ) y escribirlo de nuevo, activando la línea de escritura  $WS(\Phi_2)$ .

Otra reducción en el tamaño de las celdas RAM dinámicas se consigue usando un circuito con un solo transistor, como el que se muestra en la *figura 11.17*. Su principio de funcionamiento es conceptualmente sencillo. Basta un transistor que facilite la entrada y salida de carga en un condensador. Durante un ciclo de escritura, el valor del dato se coloca en la línea de datos ( $BL$ ) y se eleva el potencial de la línea de palabra ( $WL$ ). Dependiendo del valor del dato, la capacidad de la celda se carga o se descarga. Antes de realizar una operación de lectura, la línea de bit se precarga a una tensión de referencia,  $V_{PRE}$ . Al activar la línea de palabra ( $WL$ ), se redistribuye la carga entre la línea de bit y la capacidad de almacenamiento. Esto produce un cambio de tensión en la línea de bit y, dependiendo del sentido del cambio, se almacena un cero ó un uno. El valor de ese incremento (o decremento) de tensión es:

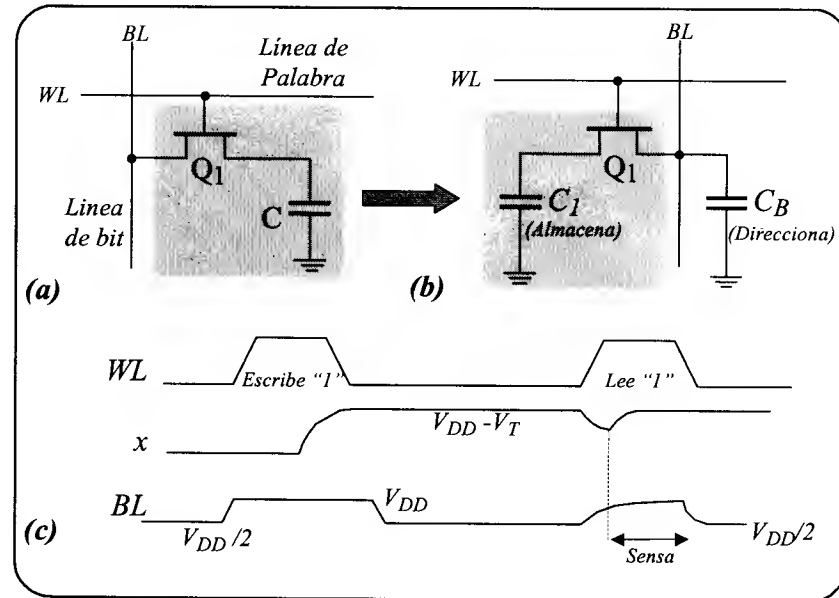
$$\Delta V = V_{BL} - V_{PRE} = (V_{BIT} - V_{PRE}) \frac{C_I}{C_I + C_{BL}} \quad \text{con} \quad C_I \ll C_{BL}$$

donde  $C_{BL}$  es la capacidad de la línea de bit,  $V_{BL}$  el potencial de la línea de bit tras la redistribución de la carga y  $V_{BIT}$  es el potencial inicial en la capacidad de almacenamiento,  $C_I$ .

Dado que la capacidad de almacenamiento,  $C_I$ , es mucho menor que  $C_{BL}$ , el cambio de tensión,  $\Delta V$ , es muy pequeño ( $\approx 250\text{mV}$ ) por lo que necesita ser adecuadamente amplificado por los sensores. Además, la lectura de estas celdas de un solo transistor es destructiva. Es decir, tras una operación de lectura, hay que restaurar el estado de la carga previamente almacenada en  $C_I$ .

Aquí nos interesa quedarnos con los aspectos cualitativos básicos del comportamiento de esta celda que vamos a repetir. El transistor  $Q_I$  está conectado a dos capacidades,  $C_I$  (almacenamiento) y  $C_{BL}$  (direccionamiento). La lectura y la escritura se realizan pasando a conducción a  $Q_I$ , mediante  $BL$ . El dato queda almacenado como un nivel de carga en  $C_I$ . Para leer, tenemos que encontrar un procedimiento de medir esa carga almacenada en  $C_I$  y para ello redistribuimos esa carga haciendo que la compartan  $C_I$  y  $C_{BL}$  (que es mucho mayor). Ahora las dos capacidades actúan como un partidor de

tensión y el incremento o decremento de potencial ( $\Delta V$ ) que ha producido esa redistribución de la carga es la medida del estado de la celda. Después lo amplificamos y restauramos el estado de carga inicial previo al proceso de lectura.



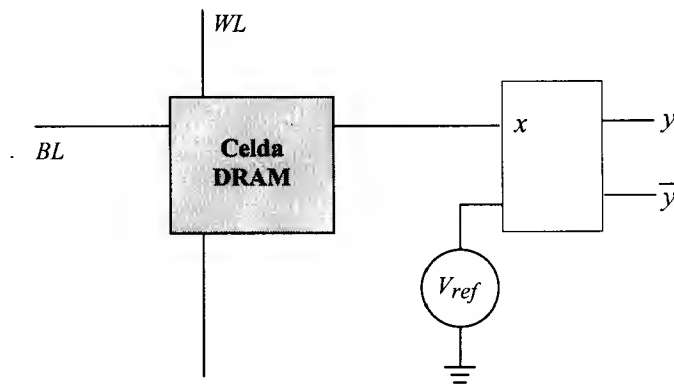
**Figura 11.17.** Celda DRAM con un solo transistor. (a) Solución mínima que provoca lectura destructiva. (b) Alternativa válida. (c) Formas de onda durante los procesos de lectura y escritura en el circuito de (b).

Mejorar la relación de transferencia de carga  $[C_I/(C_I + C_{BL})]$ , resolver la necesidad de mecanismos de refrescos y restauración del estado leído y aumentar la densidad de integración son objetivos tecnológicos cuya descripción queda fuera del alcance de este tema<sup>(\*)</sup>.

En estas celdas de RAM dinámicas la salida es única por lo que no pueden usarse directamente amplificadores sensores diferenciales como los usados en las celdas RAM estáticas. Podemos sin embargo, convertir una salida única en una salida diferencial usando una tensión de referencia ( $V_{ref}$ ) como la otra entrada diferencial ya que de hecho lo que nos interesa es convertir un pequeño pulso de

(\*) Ver por ejemplo el texto de J.M. Rabaey (Digital Integrated Circuits) incluido en las referencias específicas de este tema.

tensión en uno de los dos niveles lógicos posible (“0” ó “1”). Por eso, todo lo que quede por encima de la referencia se interpreta como un “1” y todo lo que quede por debajo, como un “0”. Es decir, el amplificador diferencial de alta ganancia opera como un comparador diferencial, tal como se muestra en la *figura 11.18*. Dependiendo del valor de  $BL$  el amplificador diferencial conmuta rápidamente hacia “1” ó hacia “0”. Hay sin embargo problemas a la hora de decidir cual es el valor apropiado para una tensión de referencia que debe de adaptarse a las variaciones que experimentan los niveles de tensión de una a otra celda. En las celdas de un transistor esto se resuelve situando al amplificador diferencial en la parte central del circuito y usando celdas adicionales con el único propósito de servir como referencia.



**Figura 11.18.** Conversión de la salida única de una celda RAM dinámica en una salida diferencial usando una tensión de referencia de valor intermedio al de los niveles lógicos “0” y “1”.

## 11.7. ORGANIZACIÓN DE LAS MEMORIAS RAM DINÁMICAS (DRAM)

Ya hemos visto que la ventaja principal de las RAM dinámicas frente a las estáticas está en la sencillez de sus celdas básicas que pueden construirse usando un solo transistor de acceso y almacenando la información a través del estado de carga de una capacidad. Esto hace que la densidad de integración sea muy alta y se alcance un equilibrio entre densidad, velocidad y coste que las hace idóneas para su uso como memoria principal de los computadores.

Sin embargo, no todo son ventajas. Al almacenar la carga en un condensador, el proceso de lectura es destructivo por lo que hay que rescribir de nuevo, antes de acabar el ciclo de lectura para restaurar el paquete de carga que había almacenado en esa celda antes de leerla. Por otro lado, existen

procesos de fuga que hacen que se degrade el contenido de las celdas aunque no hayan sido leídas, por lo que es necesario añadir mecanismos de “refresco” que restauren la carga almacenada cada cierto tiempo. Desafortunadamente, no podemos confiar en la presencia de una operación de lectura durante ese intervalo, lo que haría innecesario el refresco ya que, tras leer, siempre se rescribe lo leído.

En resumen, las DRAM ofrecen mayor capacidad para una misma área de silicio pero son más lentas que las SRAM porque, como consecuencia del tipo de celda básica que usan, tenemos que aumentar la complejidad de los circuitos adicionales necesarios para su organización. En particular la restauración tras la lectura destructiva y el refresco, para compensar las fugas.

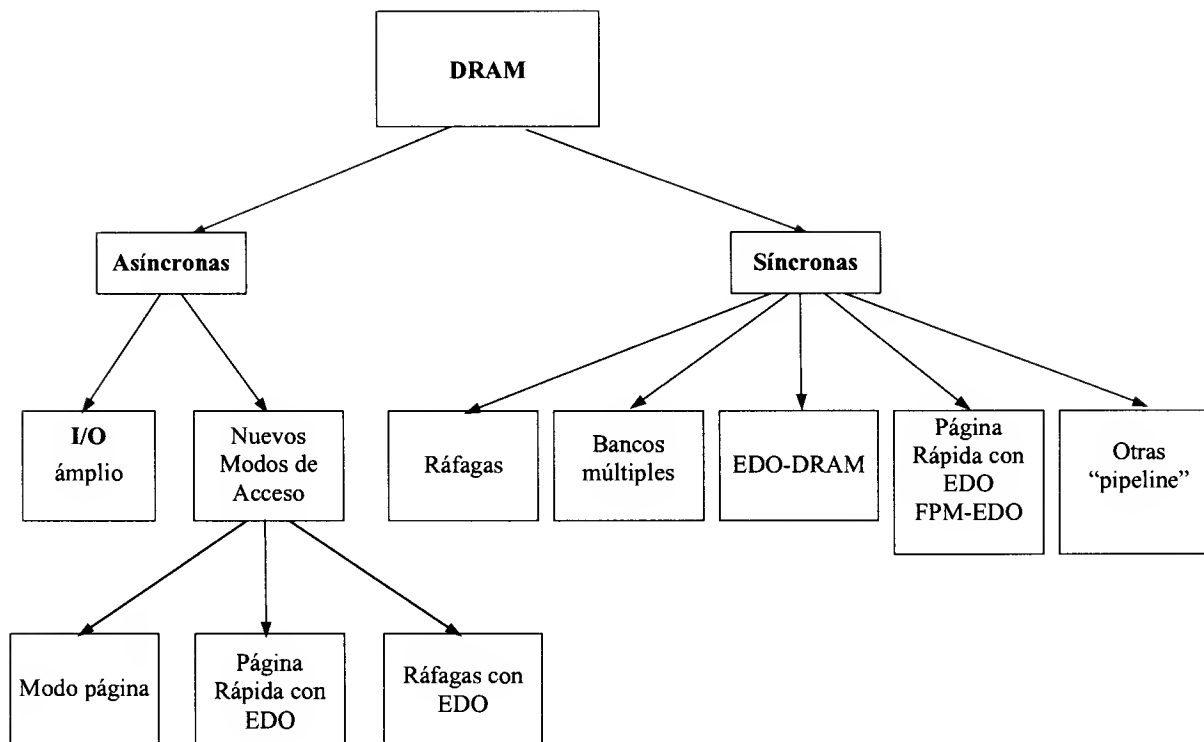
Otras modificaciones en la organización de las DRAM proceden del multiplexado de las direcciones, introducido para dividir por dos el número de terminales necesarios para la dirección. De nuevo, esto disminuye el tamaño del circuito pero incrementa el valor del tiempo de acceso haciéndola más lenta.

Hemos estructurado la presentación de este apartado en los siguientes puntos:

1. Descripción simplificada de la organización de una DRAM básica de 4Mx1.
2. Mejoras en la velocidad de las DRAM (más líneas I/O, nuevos modos de acceso, nuevas arquitecturas, ...).

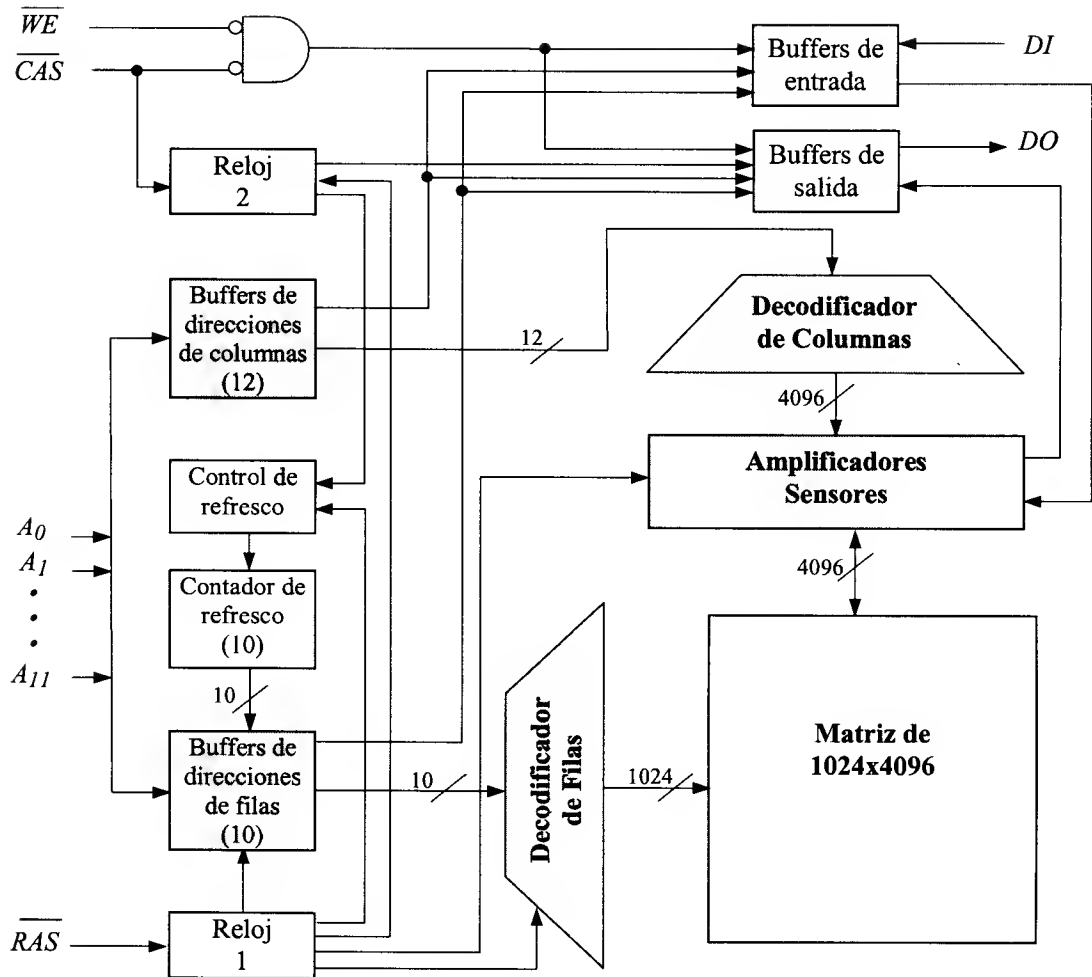
La *figura 11.19* ilustra este panorama. Algunas de las mejoras son muy análogas a las descritas previamente para las SRAM. Por ejemplo, la distinción entre comportamiento asíncrono y síncrono (con registros de bajo control del reloj central) o el significado del modo ráfaga, por lo que no volveremos a insistir aquí. Haremos sin embargo mención a los otros mecanismos, aunque de una manera muy general y cualitativa dada la extensión y complejidad del tema.

La *figura 11.20* muestra una versión simplificada del diagrama de bloques de una memoria DRAM de 4 Mx1bit, en la que se observa un núcleo común con las SRAMs vistas anteriormente y algunos circuitos adicionales necesarios para realizar las nuevas funciones de multiplexado de la dirección y refresco. Es común la matriz central de 1024x4096 celdas, los amplificadores sensores y los circuitos decodificadores de filas y columnas. Es nuevo el multiplexado de los bits de dirección, los dos relojes  $\overline{CAS}$  (Column Address Strobe) y  $\overline{RAS}$  (Row Address Strobe), el cronograma de la señal de control de lectura/escritura ( $\overline{WE}$ ) y el mecanismo de refresco.



**Figura 11.19.** Panorama de los distintos tipos de memorias DRAM. La primera distinción es entre comportamiento asíncrono o síncrono. Después, en ambos casos se busca una mayor velocidad, ampliando el bus y modificando los modos de direccionamiento..

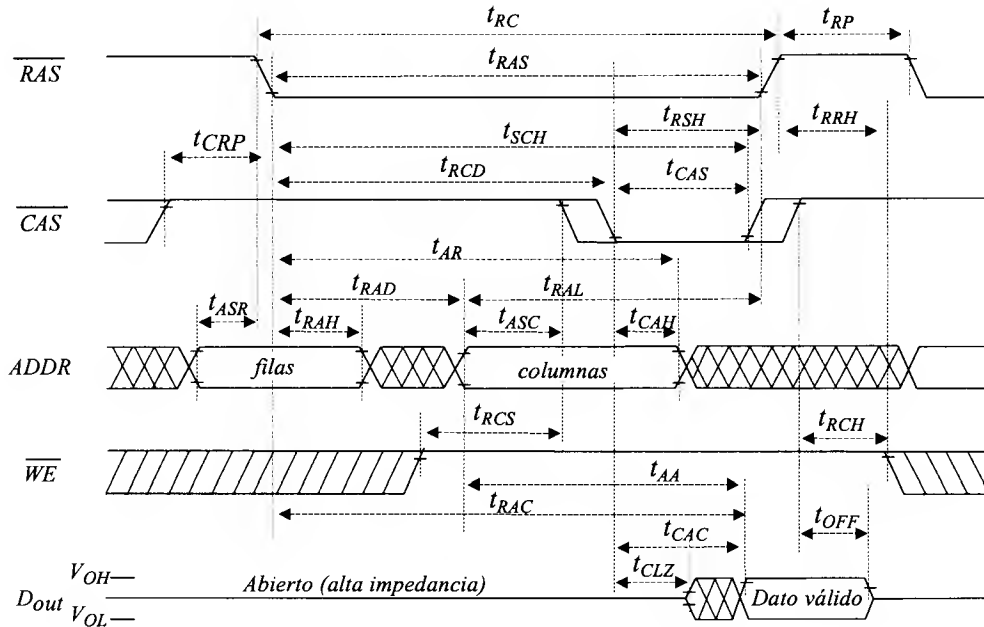
Comentaremos estas componentes nuevas. El reloj  $\overline{RAS}$  controla la selección de filas en la DRAM. La fila seleccionada está determinada por los datos presentes en las líneas de direcciones cuando la señal  $\overline{RAS}$  está activa (en baja). El reloj  $\overline{CAS}$  controla la selección de columnas en la DRAM. La columna seleccionada está determinada por los datos presentes en las líneas de direcciones cuando la señal  $\overline{CAS}$  está activa (en baja). Los relojes  $\overline{RAS}$  y  $\overline{CAS}$  actúan de forma alternativa. Primero actúa  $\overline{RAS}$  almacenando la dirección de la fila en el buffer de filas, después actúa  $\overline{CAS}$  almacenando la dirección de columna y así sucesivamente. El control  $\overline{WE}$  controla el ciclo de lectura/escritura.



**Figura 11.20.** Diagrama de bloques de una memoria DRAM de 4Mx1

Veamos ahora la evolución temporal del proceso sobre el cronograma de la *figura 11.21*. Los bits de dirección se multiplexan al comienzo de cada ciclo de lectura y se dirigen a buffers separados para filas y columnas donde son decodificados y facilitados secuencialmente por los relojes  $\overline{RAS}$  y  $\overline{CAS}$ . Hay también un contador que va seleccionando en cada intervalo de tiempo cuales son las direcciones de las celdas cuyo contenido hay que restaurar. Los amplificadores sensores realizan las dos operaciones de leer y restaurar los datos en las celdas leídas antes de finalizar cada ciclo de lectura. Para ello, los datos leídos en toda una fila se retienen hasta que se realiza esa escritura.

Como la memoria es de 4Mx1, hacen falta 22 bits para direccionar los 4M. Si seguimos la evolución del tiempo desde que se presentan los primeros 10 bits de fila en los terminales, vemos que hay que retenerlos un tiempo ( $t_{RAH}$ ) hasta que se estabilizan. Después se fijan en el buffer a la caída de  $\overline{RAS}$  quien, a su vez, activa a los amplificadores sensores. Las direcciones deben mantenerse un tiempo ( $t_{RAH}$ ), hasta que se establezca  $\overline{RAS}$ .



**Figura 11.21.** Cronograma de un ciclo de lectura en memorias DRAM

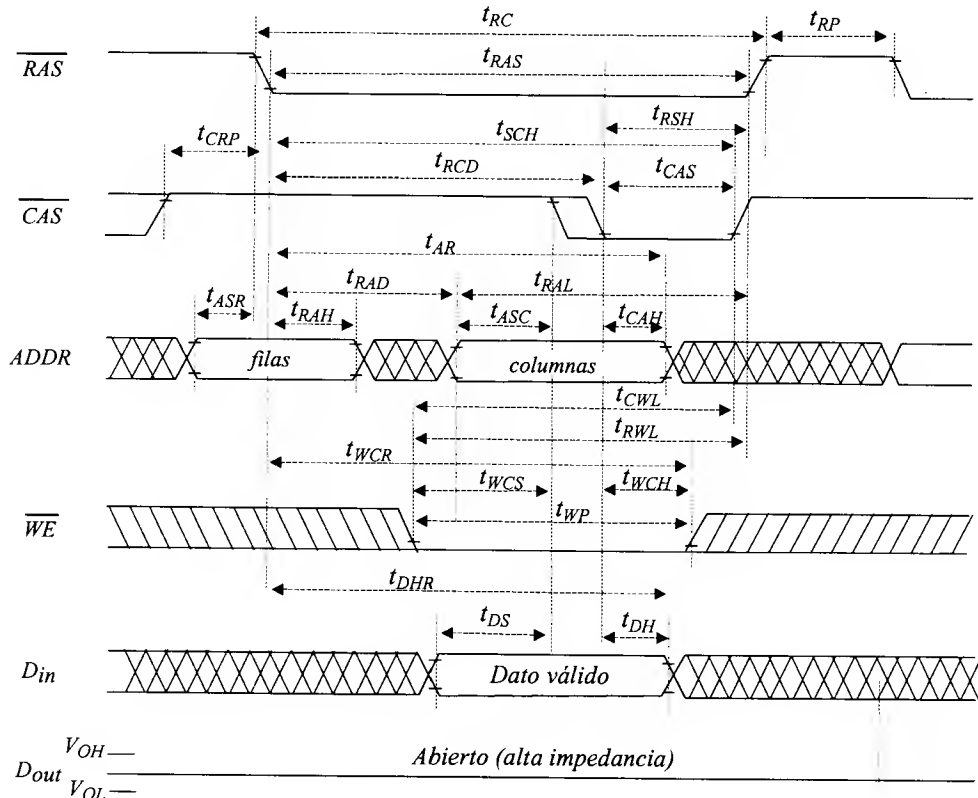
La fila seleccionada (Word Line), de un conjunto de 1024 filas, se lleva a alta y los datos en las 4096 celdas que están sobre filas se llevan a las líneas de bit (columnas) y se registran en los 4096 amplificadores sensores.

Los 12 bits de direccionamiento de columnas se estabilizan, tras un cierto tiempo  $t_{ASC}$ , y se cargan sobre el buffer de direccionamiento de columnas a la caída de  $\overline{CAS}$ , que también activa el buffer de salida y necesita un cierto tiempo,  $t_{CAH}$ , para estabilizarse. Ahora el decodificador de direcciones de columnas selecciona un bit de 4096 y lo sitúa en el buffer de salida. Los buffers de las direcciones de columnas se activan también con el flanco de bajada de  $\overline{RAS}$ , de forma que actúan como buffer de paso siempre que  $\overline{CAS}$  permanezca en alta. Esto permite que la recuperación de datos

se inicie tan pronto como la dirección de columnas sea válida, sin esperar la caída de  $\overline{CAS}$ .

Si recorremos ahora de izquierda a derecha el cronograma del ciclo de lectura vemos el conjunto de retardos que limitan la velocidad máxima de una memoria DRAM (equivalente al inverso de los valores mínimos de estos retardos), que incluyen el tiempo de asentamiento de las direcciones de filas ( $t_{ASR}$ ), los tiempos de transición ( $t_T$ ), el tiempo de retención de las direcciones de filas ( $t_{RAH}$ ), el retardo entre filas y columnas ( $t_{RCD}$ ), el tiempo de asentamiento de las direcciones de columnas ( $t_{ASC}$ ), el tiempo de asentamiento de comando de lectura ( $t_{RCS}$ ) y el tiempo de retención de las direcciones de columnas ( $t_{CAH}$ ). Finalmente hay que añadir los tiempos necesarios para la precarga de las líneas  $\overline{CAS}$  ( $t_{CRP}$ ) y  $\overline{RAS}$  ( $t_{RP}$ ), necesarios antes de iniciar un nuevo ciclo.

La operación de *escritura*, cuyo cronograma se muestra en la figura 11.22, es similar a la de lectura.



**Figura 11.22.** Cronograma de un ciclo de escritura en memorias DRAM



Las señales  $\overline{RAS}$  y  $\overline{CAS}$  fijan las direcciones de fila y columnas donde hay que escribir, el control  $\overline{WE}$  pasa a baja y se le deja que establezca su valor (retardo  $t_{WCS}$ ) y, finalmente, los datos válidos se presentan en las líneas de salida, donde se estabilizan tras un tiempo  $t_{DS}$ , antes de la bajada de  $\overline{CAS}$ . Los datos se mantienen en las líneas de entrada durante un tiempo mientras que se captan por los amplificadores sensores seleccionados por el decodificador de columnas y pasan después a las posiciones correspondientes de la DRAM, durante la fase de restauración.

Del análisis de los cronogramas de las *figuras 11.21 y 11.22* se desprenden las causas que limitan la velocidad de las DRAM y las posibles soluciones. Hay retardos que son intrínsecos al concepto DRAM y son difíciles de mejorar. Otros, tienen que ver con la arquitectura, el carácter asíncrono o síncrono del proceso con el ancho de banda de los canales de entrada y salida, etc. y son más asequibles.

### **11.7.1. Ampliación del Número de Líneas de Entrada/Salida**

Una primera forma de aumentar la velocidad de una DRAM es conseguir acceder a un mayor número de bits de datos en cada ciclo de lectura o escritura. Esto conlleva dividir la arquitectura interna por zonas o segmentos a los que se accede simultáneamente cargando en paralelo distintos grupos de amplificadores sensores. Por ejemplo, una DRAM de 1Mx16 se divide en 16 matrices de 1024x1024 bits cada una y, por consiguiente, con 1024 amplificadores sensores asociados a sus 1024 columnas. Así, durante un ciclo de lectura están activos los 16 grupos de amplificadores sensores y tras la bajada del reloj  $\overline{RAS}$ , cada uno de estos bancos de sensores posee los datos procedentes de las filas de 1024 bits direccionadas en sus respectivas matrices. Cuando el decodificador de columnas selecciona la columna direccionada, los bits procedentes de esta columna en cada una de las 16 matrices se desplazan por caminos internos paralelos entre los amplificadores sensores y los terminales de entrada/salida, ahorrando así el tiempo necesario en un acceso paralelo. Ahora tenemos 16 bits en el mismo tiempo que antes necesitábamos para obtener 1bit.

### **11.7.2. Modificaciones en los Modos de Acceso**

En el *modo de acceso aleatorio* que hemos visto, tras la activación del reloj de filas ( $\overline{RAS}$ ), había un gran número de bits que tenían que esperar en los amplificadores sensores a que su columna correspondiente fuera seleccionada. En cambio, para aquellas aplicaciones que admiten una

segmentación del conjunto de direcciones de memoria, puede usarse un direccionamiento más rápido, llamado en *modo página* en el que la línea de facilitación de la dirección de una fila ( $\overline{RAS}$ ) se mantiene activa (en baja) mientras se leen todas las columnas de esa página. Es decir,  $\overline{RAS}$ , baja sólo una vez por página y  $\overline{CAS}$ , va bajando y subiendo para cada una de las columnas de esa página.

Este modo de acceso se mejora si los buffers de direcciones de columnas se activan con la bajada de  $\overline{RAS}$  y *permanecen abiertos* cuando  $\overline{CAS}$  está en alta, dispuestos a captar las direcciones de las distintas columnas de la página en cada una de las bajadas de  $\overline{CAS}$ . Esto significa que las direcciones son activas tan pronto como la dirección de una columna es válida, sin tener que esperar a las transiciones de bajada de  $\overline{CAS}$ .

Este “*modo página rápido*” se vuelve a mejorar con la facilidad adicional  $\overline{EDO}$  (Extended Data Out). Esta extensión de la salida de datos se consigue cambiando la ruta de salida de los datos de forma que el flanco de subida del reloj  $\overline{CAS}$  ya no tenga que inhibir a las buffers de salida. Ahora los datos anteriores permanecen válidos cuando  $\overline{CAS}$  pasa a alta y se puede acceder a la siguiente dirección de columna antes de que el procesador acepte los datos válidos actuales. Es decir, eliminamos el retardo  $t_{OFF}$  de  $\overline{CAS}$ .

Dado que la mayoría de los procesadores aceptan el acceso por ráfaga, tal como lo vimos para las RAM estáticas, también se ha desarrollado este modo de acceso para las EDO DRAMs (*modo ráfaga*).

Recordemos que en el acceso por ráfagas, hay un grupo de datos a los que se accede sin necesidad de un nuevo direccionamiento externo puesto que sus direcciones son generadas internamente con un contador, a partir de los últimos bits de la primera dirección de la ráfaga. Así, sólo necesitamos controlar con  $\overline{CAS}$  y  $\overline{RAS}$  una única fila y columna, el resto de las direcciones de las columnas incluidas en la ráfaga se generan internamente.

### 11.7.3. DRAMs Síncronas con Bancos Múltiples

A pesar de todas las mejoras comentadas anteriormente para las DRAMs asíncronas, todavía hay una diferencia significativa entre la velocidad a la que los procesadores necesitan los datos que solicitan a las DRAMs y la velocidad a la que estas DRAMs se los pueden suministrar. Esta disparidad ha motivado el que se busque nuevas arquitecturas de interfaz síncrona para las DRAMs.

El proceso es análogo al que vimos en SRAM. Se trata de dotar de *registros* a las entradas de direcciones, datos y señales  $\overline{CAS}$  y  $\overline{RAS}$  y dejar el control en manos del reloj principal del procesador. Se eliminan así los “estados de espera” del procesador, mientras la DRAM realiza sus procesos intrínsecos de lectura/escritura. Ahora, ese tiempo de espera puede ser usado por el procesador para realizar otras tareas. Tras un número preestablecido de ciclos de reloj los datos estarán disponibles en la salida de la DRAM.

Siguiendo aquí también una estrategia análoga a la seguida por las SRAM, las DRAM síncronas usan también una arquitectura dividida, con múltiples zonas internas independientes que permiten aumentar su velocidad al hacer posible la lectura o escritura en una zona mientras que, de forma concurrente, en otras zonas se está procediendo a un proceso de precarga o de refresco.

## 11.8. CIRCUITOS DE MEMORIA ASOCIATIVA (CAM)

Hemos visto los circuitos de memoria SRAM y DRAM en los que todas las celdas eran direccionables con un tiempo de acceso único. Además, quedaba clara la distinción entre dirección y contenido. Primero se seleccionaba una dirección y después de leía o escribía un dato en esa dirección.

Ahora vamos a estudiar otro tipo de memorias llamadas asociativas o direccionables por contenido, CAM (Content Addressable Memory), en las que se usa el dato (el contenido) para seleccionar el conjunto de direcciones que contienen ese dato de forma completa o parcial.

Estos circuitos de memoria se usan en todas aquellas aplicaciones en las que el esfuerzo computacional más importante en todas las instrucciones que hacen referencia a memoria es un proceso de búsqueda de conjuntos de datos que coinciden con un patrón. Por ejemplo, en el diseño de memorias caché asociativas, en procesos de búsqueda rápida de datos, listas o configuraciones en máquinas orientadas a grandes bases de datos, en reconocimiento de caracteres, en redes de computadores o en redes de comunicaciones, en arquitecturas avanzadas de cálculo paralelo y en computación orientada hacia la Inteligencia Artificial aplicada. Los circuitos CAM proporcionan las ventajas de eficacia y rapidez propias del diseño paralelo a nivel electrónico (hardware) en comparación con las soluciones programadas de algoritmos de búsqueda en árboles, por ejemplo. Al comparar a nivel físico y de forma simultánea la información deseada con la relación completa de

todas las entidades previamente almacenadas, el tiempo de búsqueda se reduce en un orden de magnitud.

Hemos estructurado el contenido de esta sección en tres subapartados:

1. Aspectos básicos de la organización de un circuito CAM.
2. Celda CAM básica en CMOS.
3. Ejemplos.

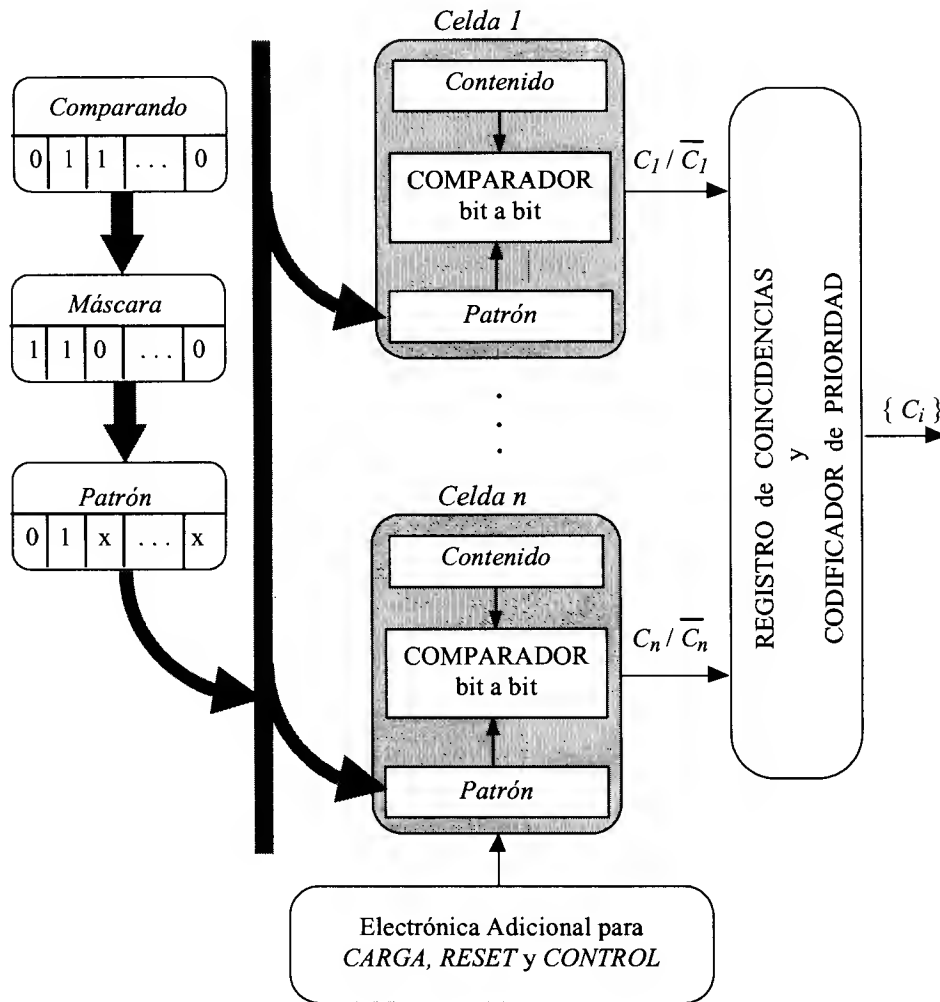
### 11.8.1. Aspectos Básicos de la Organización de un Circuito CAM

La estructura básica de un circuito de memoria CAM es la que se ilustra en la *figura 11.23*. Hay un núcleo central de almacenamiento compuesto por celdas SRAM como las que hemos visto previamente en las que a cada celda se le han añadido un cierto número de transistores para llevar a cabo la comparación entre su contenido y el del bit correspondiente de un registro adicional que contiene el *patrón* de comparación. Este patrón se obtiene a su vez de filtrar la palabra externa que queremos saber dónde está almacenada, llamada *comparando*, con una máscara que selecciona cuál es el subcampo de esa palabra (clave de búsqueda) en el que estamos realmente interesados, por ejemplo, para extraer como información el resto de los subcampos.

Así pues, en el proceso de comparación intervienen los contenidos de tres registros: *comparando*, *máscara* y *patrón* o clave de búsqueda. Esta palabra patrón o clave de búsqueda se distribuye para ser comparada en paralelo con cada una de las  $n$  palabras de la CAM. Para realizar esta comparación cada celda de la CAM posee un registro con el dato propio, otro para recibir el bit correspondiente del patrón y un circuito combinacional que detecta la coincidencia o no coincidencia entre ambos bits. Es decir, un circuito NOR exclusivo cuya salida sólo estará en alta cuando ambas entradas coinciden  $(AB + \overline{A}\overline{B})$ .

Cuando se han realizado todas las comparaciones el resultado se lleva a un registro que indica las comparaciones que han sido válidas (registro indicador o de marca) y existen tres posibilidades: que no hayamos encontrado ninguna coincidencia, que hayamos encontrado sólo una o que existan varias. Si no existe ninguna, no hay direcciones válidas para ese patrón. Si hay sólo una, se activa su bit correspondiente y si hay varias se usa un codificador con prioridad, ofreciéndose por ejemplo la dirección cuyo decimal equivalente sea el menor. Como este registro indicador de direcciones válidas debe de contener el resultado de las comparaciones de la palabra patrón con todas las de la CAM,

necesita  $n$  bits, tantos como direcciones. Hay finalmente un registro de salidas que saca las palabras seleccionadas.

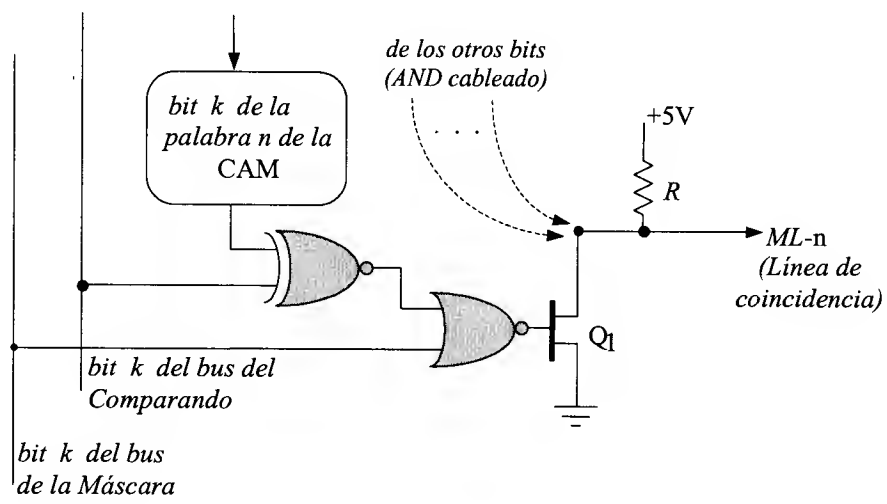


**Figura 11.23.** Esquema básico de un circuito de memoria CAM

Evidentemente, el circuito necesita electrónica adicional para controlar el proceso de comparación, estableciendo las condiciones iniciales en el registro indicador, cargando el comparando, radiando el patrón, etc. Por ejemplo, si se parte de un estado inicial en el que todos los bits del registro indicador están en alta, después de realizar la comparación, pueden quedar en alta los

bits correspondientes a las palabras cuyos contenidos han coincidido con el patrón y pasar a baja los bits correspondientes a las que no han coincidido.

La *figura 11.24* muestra el esquema elemental de comparación, primero para un bit y después con el AND cableado para los  $n$  bits de cada palabra. Este es el circuito que usa la CAM CMOS Am99C10A cuyo diagrama de bloques se muestra en la *figura 11.25* y el significado funcional de sus terminales en la *11.26*.



Bit $k$ en CAM	Bit $k$ en Comparando	Salida XNOR	Bit $k$ en Máscara	Salida NOR	
0	0	1	0	0	<i>match</i>
0	1	0	1	0	<i>match</i>
1	0	0	0	1	-----
1	1	1	1	0	<i>match</i>

**Figura 11.24.** Comparador de un bit de la CAM Am99C10A

Este circuito de memoria CAM tiene una capacidad de 256 palabras de 48 bits, y es programable por el usuario. Cada una de las 256 celdas CAM incluye un registro, un comparador que genera un bit de coincidencia (*match flag*) y otros 8 bits con la dirección de la palabra que coincide

con el patrón. Hay pues 256 líneas de coincidencia ( $ML0$ , ...,  $ML255$ ). El comparador de bit es un circuito coincidencia (XNOR), tal como comentamos anteriormente. Tras esta comparación se pasa por una puerta NOR con el bit correspondiente de la máscara y el resultado de la comparación de bit se lleva a un transistor ( $Q_1$ ) que realiza el AND cableado (drenador abierto) de los 48 bits de la palabra (*figura 11.24*).

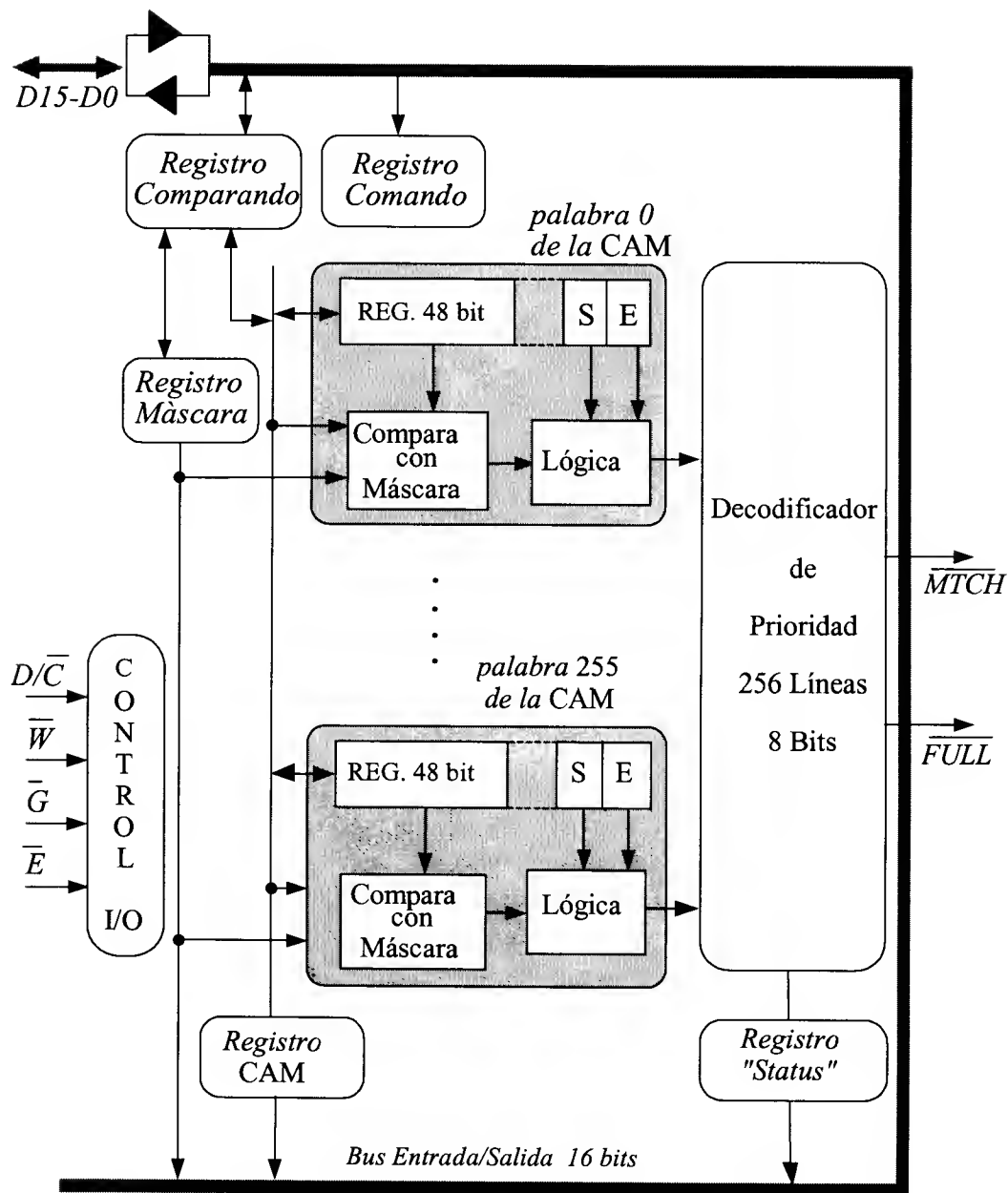
Cuando el bit de la CAM AM99C10A coincide con el del comparando, la salida de XNOR está en alta. Después se ve si interesa o no esa comparación calculando el NOR con el bit de la máscara. La salida de la puerta NOR (que realiza la función  $\overline{A \cdot B}$ ) está en baja cuando hay coincidencia y esto ocurre en tres circunstancias: Cuando los bits coinciden, independientemente del estado de la máscara (00,11) y cuando no coinciden pero la máscara nos dice que ese bit no es relevante en la comparación (0,1). La señal  $ML$  permanece en alta para esa palabra si los 48 transistores  $Q_1$  dicen que sus NOR correspondientes están en baja.

Si nos fijamos ahora en el diagrama de bloques de la *figura 11.25* vemos cómo se integra esta operación de comparación bit a bit ilustrada en la *figura 11.24* con el funcionamiento global de la CAM. Cuando se detecta una coincidencia el codificador de prioridad identifica la dirección o direcciones correspondientes a estas coincidencias (líneas  $ML$  en alta). Si hay al menos una  $ML$  en alta el codificador activa la línea  $\overline{MTCH}$  y transmite al registro “status” los 8 bits de la dirección más baja en la que se ha detectado una coincidencia. Las otras coincidencias pueden recuperarse a partir de dos bits adicionales que poseen todas las palabras de la CAM.

Esta CAM es programable existiendo un conjunto de códigos de operación para inicializarla, cargar los registros de comparando y máscara, desplazar los contenidos de estos registros a cada una de las posiciones de memoria (carga de la CAM), leer el contenido de la CAM mediante la transferencia de los contenidos de sus 256 direcciones al registro de comparandos, etc...

El procedimiento normal de uso es cargarla con los contenidos deseados, cargar el registro máscara y pasar al modo de comparación, leyéndose en el registro “status” el valor del bit de coincidencia (*match flag*) y la dirección donde está el dato coincidente. Las coincidencias múltiples se leen de forma sucesiva empezando por la de dirección más baja.

Más adelante, tras el estudio de la celda básica, volveremos con otros ejemplos de circuitos CAM comerciales representativos del estado actual de la tecnología electrónica en este campo.



**Figura 11.25.** Diagrama de bloques funcionales de la memoria CAM Am99C10A, usado para ilustrar la arquitectura b sica de este tipo de circuitos (S=Skip bit, E=Empty bit).



$D/\overline{C}$	Selección de modo (Datos o Comandos)
$\overline{W}$	Facilitación de la escritura en los registros internos y en las celdas de la CAM
$\overline{G}$	Facilitación de la lectura en los registros de salida
$\overline{E}$	Selección de circuito
$D15-0$	Bus de datos. Es un bus de 16 bits, bidireccional y de tres estados
$\overline{FULL}$	Indica que todas las celdas han sido cargadas
$\overline{MTCH}$	Indicador de coincidencia en una o más palabras

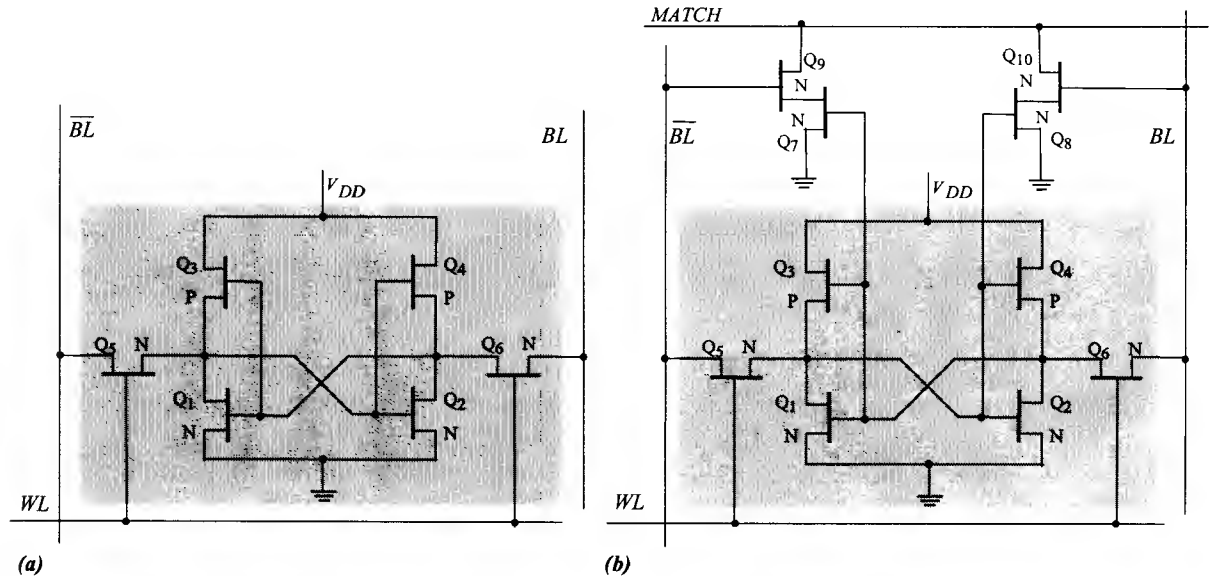
**Figura 11.26.** Significado funcional de los terminales de la memoria CAM Am99C10A.

### 11.8.2. Celda CAM básica en CMOS

Ya conocemos los circuitos correspondientes a celdas RAM estáticas en MOS. Las celdas en CMOS son análogas (dos inversores acoplados) sólo que en este caso los inversores son CMOS en vez de ser NMOS ó PMOS. En la parte (a) de la *figura 11.27* se muestra el circuito correspondiente a una celda SRAM de 6 transistores, cuatro para la síntesis de los dos inversores acoplados y dos para leer y escribir. En la parte (b) de la *figura 11.27* se muestra la celda CAM correspondiente en la que se han añadido cuatro transistores más para realizar las funciones de comparación del estado de alta o baja del bit almacenado en los otros seis con el bit correspondiente del registro que almacena al comparando o al patrón, tras pasar el comparando por la máscara. Es decir, para realizar la función NOR exclusivo.

Para leer y escribir en esta celda en modo RAM ya conocemos el mecanismo. Para comparar, la línea de “match” se precarga a alta y las líneas de bit se llevan al nivel correspondiente al bit almacenado en el registro comparando. Como la línea de palabra no está activada, el estado del biestable formado por los seis transistores SRAM no queda afectado. En cambio, los transistores que sintetizan el NOR exclusivo realizan la comparación del estado almacenado con los niveles de bit del comparando. El estado almacenado está representado en las tensiones de puerta de los pares  $Q_1$ - $Q_3$  y

Q<sub>2</sub>-Q<sub>4</sub>. Como estos terminales de puerta están unidos a las puertas de Q<sub>7</sub> y Q<sub>8</sub>, respectivamente, los circuitos de comparación (pares Q<sub>7</sub>-Q<sub>9</sub> y Q<sub>8</sub>-Q<sub>10</sub>) “conocen” el estado almacenado en la parte RAM de la celda CAM. Por otra parte las puertas de los transistores Q<sub>9</sub> y Q<sub>10</sub> están activadas por los niveles de tensión correspondientes al bit del comparando. Si estos dos niveles no son el mismo, la línea de “match” (drenadores de Q<sub>9</sub> y Q<sub>10</sub>) pasa a baja, indicando que no se ha encontrado coincidencia. Todos los comparadores de bit de una misma palabra se conectan a la misma línea de “match”, de forma que basta con que falle una comparación para que esa línea pase a baja. Sólo cuando todos los bits de una palabra han producido comparaciones positivas, la línea de “match” de esa palabra pasa a alta. Es decir, al igual que vimos en la *figura 11.24*, se está realizando el AND cableado del resultado de la comparación de los bits individuales.



**Figura 11.27.** (a) Celda RAM estática. (b) Celda CAM obtenida a partir de la RAM añadiéndole cuatro transistores para realizar la función NOR exclusivo entre su contenido y el bit correspondiente del comparando.

### 11.8.3. Ejemplos de circuitos CAM

Como siempre, después de conocer los aspectos básicos de la organización de un tipo de circuito y la conexión con el nivel de su síntesis con transistores, creemos conveniente tomar contacto

con el estado actual de la tecnología electrónica describiendo algún circuito real representativo del campo. En este caso hemos seleccionado la familia LANCAM B de Music Semiconductors, que produce circuitos CAM en tecnología CMOS organizados en base a palabras de 64 bits (así opera en el bus interno), con un ancho de banda de 16 bits para entrada y salida, velocidad de comparación de  $50ns$  y otras características tales como la posibilidad de segmentar el contenido en partes RAM y partes CAM y la inclusión de registros con capacidad de desplazamiento de máscaras y comparandos, permitiendo así la implementación hardware de algoritmos de búsqueda asociativa de prioridad. La capacidad de estos circuitos CAM puede ser de 1K, 4K ó 8K, siendo además ampliable esta profundidad mediante la conexión en cascada.

Desde el punto de vista operacional estos circuitos LANCAM siguen el procedimiento descrito en el apartado 11.8.1. Los datos cargados en el registro comparando se comparan automáticamente con los contenidos de todas las direcciones CAM válidas. El circuito indica si una o más de estas direcciones válidas contienen datos coincidentes con los que buscamos. Cada posición CAM puede estar en uno de los cuatro estados siguientes: (*válida*, *vacía*, *omitida* y *RAM*). Estas cuatro condiciones están codificadas en los dos últimos bits (28 y 29) del *registro de estado*. Con la carga de los registros de control, comparando y máscara se inicia el proceso de comparación ofreciendo los resultados de la búsqueda tras  $50ns$  y después el registro status informa sobre esos resultados, incluyendo todos los flags y las direcciones en las que se ha tenido éxito al comparar. El circuito incluye dos *registros de máscara* y cuatro señales de control que entran al decodificador de instrucciones para facilitar la gestión de la CAM de forma flexible y con una carga mínima sobre la programación externa. La *figura 11.28* muestra una versión simplificada del diagrama de bloques.

Los terminales *DQ15-0* son las líneas del bus externo que transportan datos, comandos y descripciones del estado de la CAM (status) entre el interior y el exterior del circuito. Las señales  $\overline{W}$  y  $\overline{CM}$  controlan el sentido del tráfico de señales y la naturaleza de esas señales. Cuando  $\overline{E}$  está en alta las líneas *DQ15-0* pasan a alta impedancia.

Al control entran cuatro señales,  $\overline{E}$ ,  $\overline{W}$ ,  $\overline{CM}$  y  $\overline{RESET}$  y una quinta,  $\overline{EC}$ , va a la lógica de coincidencia. La primera señal ( $\overline{E}$ ), es la entrada de facilitación del circuito. Su flanco de bajada registra las otras señales de control ( $\overline{W}$ ,  $\overline{CM}$  y  $\overline{EC}$ ). La señal  $\overline{W}$  (*write enable*) selecciona entre lectura y escritura y  $\overline{CM}$  nos dice si las señales en las entradas *DQ15-0* deben interpretarse como comandos ( $\overline{CM}$  en baja) o como datos ( $\overline{CM}$  en alta), dando lugar a cuatro tipos de ciclos:

$\overline{W}$	$\overline{CM}$	Tipo de ciclo
L	L	<i>Escritura de comandos</i>
L	H	<i>Escritura de datos</i>
H	L	<i>Lectura de comandos</i>
H	H	<i>Lectura de datos</i>

Finalmente, la señal  $\overline{EC}$  facilita que el bit de marca  $\overline{MF}$  muestre el resultado de una comparación. Si  $\overline{EC}$  está en baja, en el flanco de bajada de  $\overline{E}$  en un determinado ciclo, se facilita la salida  $\overline{MF}$ . Esta señal de salida pasa a baja cuando han ocurrido una o más coincidencias en el proceso de comparación.

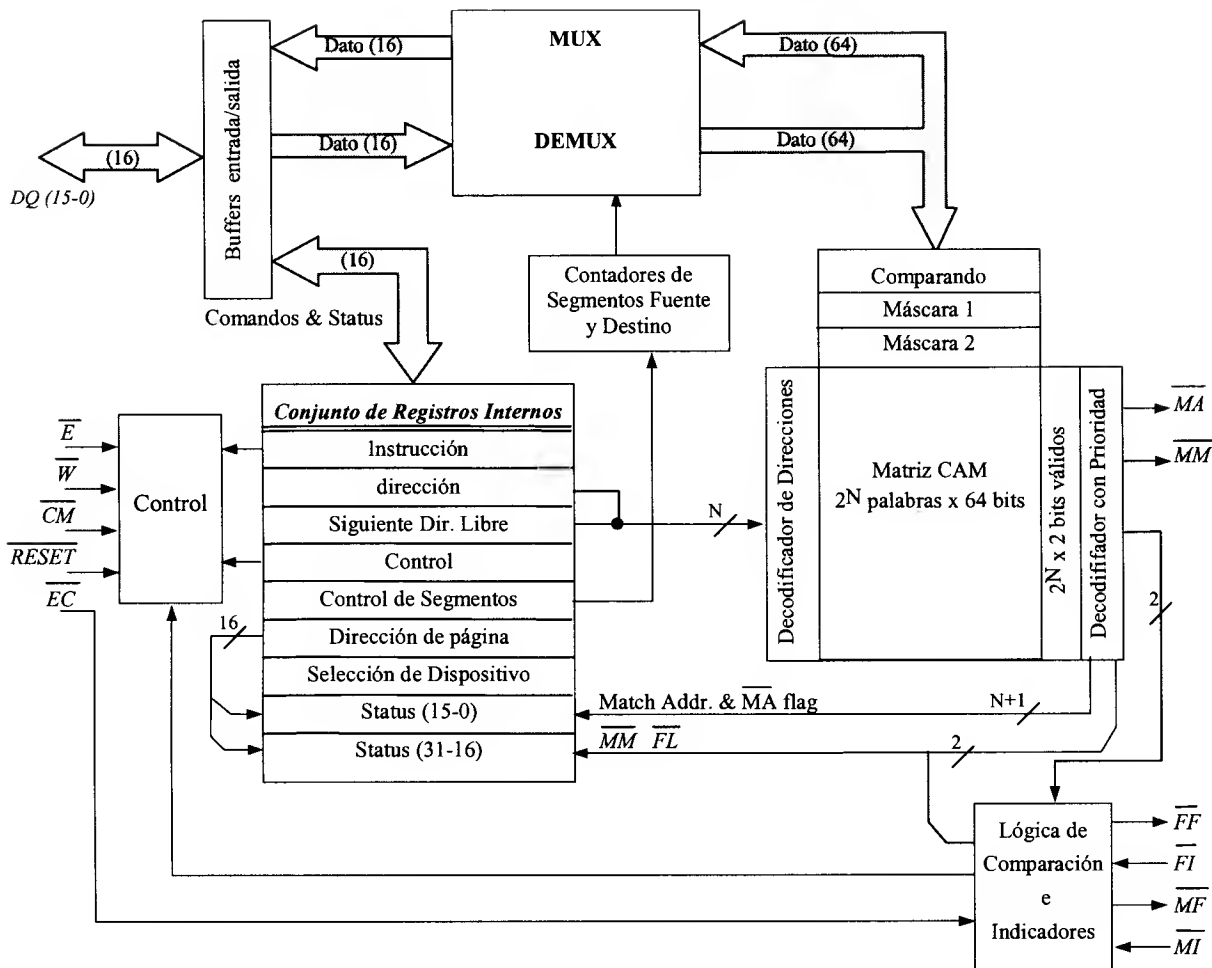
Hay un conjunto de señales como  $\overline{MI}$  (*match input*),  $\overline{MA}$  (*device match flag*),  $\overline{MM}$  (*device múltiple match flag*),  $\overline{FF}$  (*full flag*) y  $\overline{FI}$  (*full input*) que sirven para conectar en cascada vertical varios circuitos CAM y para añadir codificadores externos de prioridad, distinguiendo en cada caso la situación en cada uno de los circuitos de la cascada de la situación global de toda la CAM. Veamos su descripción funcional.

La entrada  $\overline{MI}$  prioriza los dispositivos en un sistema CAM que conecta varios en cascada. Para realizar esta función, el terminal  $\overline{MI}$  del primer circuito de la cascada se conecta a la alimentación (alta). En el resto, se conectan a las salidas  $\overline{MF}$  de los circuitos previos.

La salida  $\overline{MA}$  está en baja cuando se ha encontrado coincidencias en el actual ciclo de la comparación o en el anterior. Esta salida no puede ser modificada por  $\overline{EC}$  ó  $\overline{MI}$  y por consiguiente refleja el valor del indicador de coincidencias de ese circuito, a partir de su registro status.

La salida  $\overline{MM}$  está en baja cuando se han producido dos o más coincidencias válidas en el ciclo actual o en el anterior.

La salida  $\overline{FF}$  pasa a baja cuando no se han encontrada direcciones vacías en ese dispositivo. El flag que indica que todo el sistema CAM está lleno es el  $\overline{FF}$  del último circuito de la cadena. De forma análoga, la entrada  $\overline{FI}$  (*full input*) genera un indicador de que el sistema está lleno cuando los circuitos CAM individuales se han conectado en cascada. Esta entrada  $\overline{FI}$  se conecta a la salida  $\overline{FF}$  del dispositivo previo.



**Figura 11.28.** Versión simplificada del diagrama de bloques de los circuitos de memoria asociativa de la familia LANCAM B de Music Semiconductors.

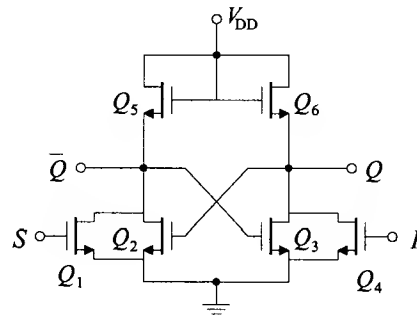
No es aconsejable profundizar más en la descripción operacional de este circuito CAM porque, al igual que comentamos en el tema 7 al hablar de memorias EEPROM tipo flash, la evolución de la tecnología electrónica nos lleva a circuitos supuestamente componentes de un sistema de cálculo digital que, a su vez, incluyen procesadores internos con un repertorio propio de instrucciones para su programación con el propósito final de ser parcialmente autónomos. Se libera así al procesador central de una parte relevante del esfuerzo de cálculo que antes tenía que dedicarse a la gestión software de

estas memorias CAM pero, en contrapartida, la complejidad del circuito CAM resultante (con su propio repertorio de instrucciones) supera las fronteras de una descripción razonable a nivel electrónico. Es decir, en términos de señales, registros y cronogramas.

## 11.9. PROBLEMAS

Los ejercicios de este capítulo se encaminan a reforzar la comprensión de estos conceptos: celdas, organización del direccionamiento y comparación de contenidos.

**E.11.1.** *En la figura se representa una celda básica de memoria formada por un biestable R-S en tecnología NMOS. Generar su tabla de verdad.*



**E.11.2.** *El circuito de la figura 11.9 muestra el esquema de una celda RAM estática con transistores bipolares multiemisor. Analiza la evolución temporal de la señal en el punto (1) (colector del transistor  $Q_1$ ) a partir del estado inicial en el que la celda almacena un “0” durante la siguiente secuencia de procesos:*

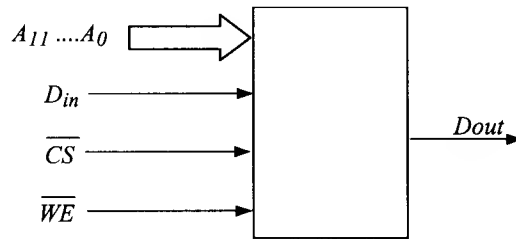
- La celda almacena un “0” y no está direccionada.*
- La celda se direcciona y se escribe un “1”*
- Sigue direccionada y se lee su estado.*
- Se escribe un “0” de nuevo.*

*Describir cualitativamente la evolución de las señales  $BL$ ,  $\overline{BL}$  y en la línea de selección de dirección correspondiente a esta secuencia.*

**E.11.3.** *El circuito de la figura 11.12 muestra el esquema de una posible celda RAM estática en tecnología MOS. Dibujar el cronograma en el que se muestre la evolución de las señales en*

$BL$ ,  $WL$  y  $\overline{BL}$  durante los procesos de lectura y escritura (a, b, c y d) que allí se describen.

- E.11.4.** Recordando que una celda RAM estática es esencialmente un biestable (dos inversores acoplados) junto con las facilidades de direccionamiento y de lectura y escritura (por ejemplo, a través de un transistor que actúa como puerta de transmisión), proponer y justificar el funcionamiento de una posible celda RAM estática en tecnología CMOS.
- E.11.5.** Supongamos que disponemos de una memoria RAM mínima consistente en sólo dos biestables R-S con “preset” y “clear”. Diseñar todos los circuitos externos para:
- Seleccionar en lectura y escritura.
  - Direccionar uno u otro biestable, ó ninguno de los dos.
  - Leer en el R-S seleccionado.
  - Escribir (0,1) en el R-S seleccionado. Es decir, sintetizar sobre esta memoria mínima y usando puertas AND, OR y NOT, todas las funciones de control que son necesarias para una memoria RAM estática.
- E.11.6.** Supongamos que disponemos del circuito de la figura que es una memoria RAM estática organizada como 4096 “palabras” de un sólo bit organizada en forma de matriz de 64 filas por 64 columnas.



- Diseñar el circuito selector de filas que pase de (A0, ..., A5) a las filas correspondientes. Tenemos así también el circuito selector de columnas.
- ¿Qué contador binario deberíamos usar para conseguir recorrer todas las direcciones posibles de forma secuencial?. ¿Cómo deberíamos conectarlo?.
- ¿Cómo podríamos usar este mismo contador para controlar las señales Din, Dout,  $\overline{CS}$  y  $\overline{WE}$ , de forma que, empezando en la dirección 0 (00...0), vaya leyendo hasta la posición 2000, escriba “unos” desde la 2001 a la 3000 y escriba “ceros” desde la 3001 a

*la 4.096?*

**E.11.7.** *Supongamos que disponemos de cuatro circuitos como los del ejercicio anterior de 4096 “palabras” de un bit.*

- a)** *¿Cómo podríamos diseñar un circuito que a través del  $\overline{CS}$  de las cuatro memorias las organizara en 4096 palabras de 4 bits?*
- b)** *¿Cómo podríamos asegurarnos después de la escritura de una determinada configuración (1100, por ejemplo) que efectivamente ese es el contenido de la celda?. (Pensar en la posibilidad de almacenar lo que se escribe, leer lo último escrito y comparar). Diseñar el circuito que realiza esta comprobación y genera un pulso sólo cuando hay error, es decir, cuando el estado de la celda direccionada no coincide con la palabra que queríamos escribir.*
- c)** *¿Cómo podríamos visualizar las palabras que se van a escribir y las que se van leyendo?.*

**E.11.8.** *Disponemos de nuevo de cuatro circuitos de memoria del tipo 2147 (4096 palabras de un bit) ¿cómo podríamos conectarlos para:*

- a)** *Conseguir una memoria de 16384 palabras de 1 bit.*
- b)** *Conseguir una memoria de 8192 palabras de 2 bits.*

*Dibujar en ambos casos las conexiones correspondientes y los circuitos externos necesarios para coordinar los niveles en las líneas de control y de datos de las cuatro memorias.*

**E.11.9.** *La figura 11.23 muestra el esquema conceptual de una memoria CAM incluyendo el “comparando”, la “máscara”, el “patrón” y el proceso paralelo de comparación con todos los contenidos. Finalmente, incluye el codificador con prioridad para ver si hay alguna coincidencia y para ordenarla en el caso de haber varias. Ya sabemos que existen soluciones integradas para todas estas funciones pero el objetivo de este ejercicio es tratar de diseñar una CAM mínima para reforzar la comprensión de los procesos de lógica combinacional y secuencial que intervienen.*

*Supongamos entonces que disponemos de una memoria CAM consistente en sólo dos palabras de dos bits, almacenadas en cuatro biestables, R-S. Diseñar los circuitos necesarios para:*

- a)** *Pasar el comparando por la máscara y generar el patrón.*



- b) *Realizar la comparación simultánea del patrón con las dos palabras de memoria.*
- c) *Generar cuatro salidas ( $O_1$ ,  $O_2$ ,  $O_3$ ,  $O_4$ ) dependiendo del resultado de la comparación:*

*$O_1$ : Si no coincide con ninguna de las dos.*

*$O_2$ : Si coincide con el contenido de una celda.*

*$O_3$ : Si coincide con el contenido de la otra celda.*

*$O_4$ : Si coincide con el contenido de ambas celdas.*

*Proponer también todo el circuito de temporización a partir de un reloj principal*

**E.11.10.** *En el diseño del ejercicio anterior sólo abordábamos los problemas distintivos de una memoria CAM, sin preocuparnos de la escritura de datos ni de la lectura, en el sentido convencional de las memorias RAM. En cambio, en el ejercicio E.11.5 ya teníamos “resuelto” el problema del direccionamiento, lectura y escritura en biestables R-S.*

*Realizar ahora una síntesis de ambos ejercicios dotando al circuito solución del ejercicio anterior de la capacidad de lectura y escritura en el sentido RAM. Es decir,*

- a) *Diseñar los circuitos necesarios para seleccionar una u otra de las dos únicas direcciones de memoria.*
- b) *Diseñar los circuitos necesarios para leer el contenido de las direcciones de las celdas seleccionadas, es decir, las correspondientes a las salidas  $O_2$  y  $O_3$  del ejercicio anterior.*



---

# ***Preparación de la Evaluación***

---

*Hemos descompuesto el tema en tres objetivos:*

Objetivos
1: SRAM
2: DRAM
3: CAM

## **1. SRAM**

*1.1. En el tema 7 estudiamos las memorias de lectura/escritura no volátiles y con posibilidad de acceso aleatorio (EEPROM). Ahora hemos estudiado las memorias de lectura/escritura, volátiles y con posibilidad de acceso aleatorio (SRAM y DRAM). ¿Podría explicar sus analogías y diferencias especificando los conceptos clave en esta clasificación?:*

▲ *Modos de lectura y escritura*

▲ *Volatilidad de los datos almacenados*

▲ *Modo de direccionamiento*

*1.2. Suponga el caso más elemental posible de memoria SRAM con sólo dos palabras de un bit cada una y suponga también que dispone de biestables J-K con preset y clear como los estudiados en el tema 8. Añádale ahora todos los circuitos adicionales para selección, direccionamiento, lectura y escritura en esta mini-RAM estática. Si tiene dificultades, revise el ejercicio E.11.5 del texto de problemas.*

*1.3. Incremente ahora un poco la complejidad de la memoria diseñando los circuitos correspondientes para una memoria SRAM de dos palabras de 2 bits.*

*1.4. ¿Podrá ahora razonar sobre la generalización de los ejercicios anteriores (1.2 y 1.3) al caso de una memoria SRAM de 512 filas x 256 columnas x 8bits análoga a la de la figura 11.4?*

**a)** *¿Cuál es la capacidad de esta memoria?*

- b) *¿Cuántos bits son necesarios para direccionar sus filas?*
- c) *¿Y para direccionar sus columnas?*
- d) *¿Qué significa el 8 final ( $A \times B \times 8$ )?*
- e) *¿Cómo pueden hacerse bidireccionales las 8 líneas de entrada/salida de datos?*
- f) *¿Cuál es el conjunto mínimo de señales de control necesarias para gestionar los ciclos de lectura/escritura?*
- g) *¿Qué significa que esta memoria SRAM sea asíncrona?*
- h) *¿Cuál es la forma cualitativa del cronograma de un ciclo de lectura?*
- i) *¿Y la de un ciclo de escritura?*
- j) *¿Cómo se puede extender esta memoria a más bits?*
- k) *¿Cómo se puede extender a más palabras del mismo número de bits?*

Estas cuestiones pretenden ser representativas de los conceptos que deben conocerse sobre la organización de una memoria SRAM y permiten un cierto grado de autoevaluación por subobjetivos.

**1.5.** *Recordará que la diferencia entre la velocidad de los procesadores y la de las memorias ha sido el motor de la evolución en la tecnología, las interfaces y la arquitectura interna de las SRAM. ¿Podría ahora seleccionar un conjunto de criterios y clasificar, de acuerdo con esos criterios, las distintas innovaciones introducidas en las memorias SRAM?*

**1.6.** *Como sabe, todas las celdas SRAM son biestables R-S a los que se han añadido las facilidades de direccionamiento y lectura/escritura. Describa el funcionamiento de la celda bipolar del ejercicio E.11.2 para la siguiente secuencia de sucesos:*

- ▲ *La celda almacena un “1” y está direccionada.*
- ▲ *La celda deja de estar direccionada.*
- ▲ *La celda se vuelve a direccionar y se escribe un “0”.*
- ▲ *La celda sigue direccionada y se lee para comprobar que efectivamente ahora su estado es “0”.*

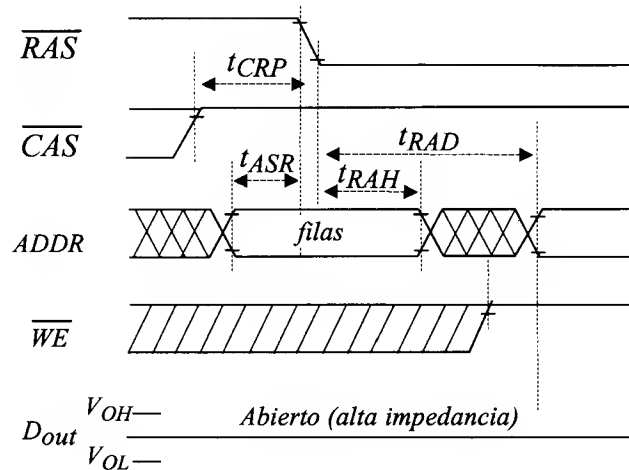
**1.7.** *¿Podría explicar el efecto que tiene la inclusión de dos diodos de barrera Schottky entre las líneas de selección de palabra y los colectores de los transistores multiemisor en la celda SRAM del ejercicio anterior? ¿Qué ventajas tiene la celda SBD SRAM resultante?*

## 2. DRAM

- 2.1. *En el texto de problemas analizamos con detalle el comportamiento de la celda SRAM de 6 transistores en tecnología MOS (ejercicio E.11.3). ¿Qué tendríamos que añadirle para convertirla en dinámica?.*
- 2.2. *¿Podría explicar la evolución de las celdas DRAM desde la de cuatro transistores hasta la de un sólo transistor?. ¿Para qué se usa realmente el transistor en este último caso?. ¿Dónde se almacena la información?. ¿Por qué es necesario un amplificador-sensor de relativamente alta ganancia?.*
- 2.3. *¿Cuáles son las ventajas de los amplificadores sensores en configuración diferencial?. ¿Por qué no pueden usarse siempre?.*
- 2.4. *Dibuje el diagrama de bloques de una hipotética DRAM y distinga claramente cuales son las partes de esa organización que se comparte con las SRAM y cuáles son las funciones adicionales propias de las DRAM.*
- 2.5. *Como recordará, la lectura en los DRAM es destructiva por lo que hay que almacenar transitoriamente el valor leído para volverlo a escribir en la misma celda antes de finalizar un ciclo de lectura.*

*Aunque ya sabemos que en los biestables J-K de las celdas estáticas la lectura no es destructiva, le sugerimos que use el circuito del ejercicio 1.2 (de una memoria con sólo dos “palabras” de un bit) para diseñar los circuitos complementarios necesarios para volver a escribir del valor leído. Si encuentra dificultades puede revisar el ejercicio E.11.7 donde se plantea un problema análogo (comprobar, después de un ciclo de escritura, que hemos escrito realmente lo que queríamos escribir).*
- 2.6. *Los relojes  $\overline{RAS}$  y  $\overline{CAS}$  son la componente más relevante del control de los ciclos de lectura y escritura en memorias DRAM. ¿Podría explicar su significado funcional y dibujar después el cronograma de un ciclo de lectura?. Use como referencia, por ejemplo la figura 11.20, correspondiente a una DRAM de 4Mx1.*
- 2.7. *Hay una lógica intrínseca a todos los cronogramas que describe la secuencia de sucesos que deben ocurrir en las distintas señales de control ( $\overline{RAS}$ ,  $\overline{CAS}$ ,  $\overline{WE}$ ) y en las líneas de datos y*

direcciones. Explíquela y trate de completar el cronograma siguiente, correspondiente a un ciclo de lectura



- 2.8. La lucha permanente en la electrónica de las memorias DRAM se centra en: (1) conseguir mantener su ventaja frente a las SRAM en cuanto a densidad de integración y, por consiguiente, su mayor capacidad para una misma área de silicio y (2) reducir a la vez las desventajas centradas esencialmente en la menor velocidad de acceso. Detrás de ambas memorias, SRAM y DRAM, siempre están los procesadores que son más rápidos.

Describe la evolución de las DRAM usando criterios de interfaz, direccionamiento, arquitectura y tecnología e intentando llegar desde las propuestas iniciales hasta las soluciones actuales (síncronas, hiperpágina, EDO, etc...). Reflexione sobre su nivel de comprensión de estas mejoras e intente dibujar un grafo de clasificación de las mismas.

### 3. CAM

- 3.1. Si mezclamos adecuadamente las soluciones de los ejercicios E.11.5 y E.11.9 no le será difícil responder al siguiente problema:

Diseñar un circuito de memoria CAM para dos palabras de dos bits usando una entrada D para direccionar, un control WE y un circuito detector de coincidencias (XNOR), con un registro de comparando de 2 bits ( $C_1$  y  $C_0$ ) y otro de máscara ( $M_1$ ,  $M_0$ ), también de 2 bits.

3.2. *Toda celda CAM puede obtenerse a partir de una RAM añadiéndole los transistores necesarios para las funciones adicionales de comparación del contenido de la celda con el patrón de comparación. En el apartado correspondiente del tema (11.8.2) hemos visto un ejemplo de extensión de una celda SRAM de 6 transistores a otra CAM con 10 transistores.*

*¿Podría explicar el funcionamiento de este circuito en el caso en el que tiene almacenado un “0” y el bit del comparando es “1”?*

*El punto en el que hay que hacer énfasis es en la comparación. La primera parte, consiste en poner el bit a comparar en las líneas de bit, pero sin activar la línea de palabra.*



## REFERENCIAS BIBLIOGRÁFICAS

1. Advanced Micro Devices: “*CMOS Memory Products*”. Data Book/Handbook. 1991.
2. Betty Prince, “*High Performance Memories*”, J. Wiley, 1999.
3. Cypress Semiconductor “*High Performance Data Book*”. 1995.
4. MUSIC Semiconductors. [www.music-ic.com](http://www.music-ic.com)
5. Rabaey, J.M: “*Digital Integrated Circuits. A Design Perspective*”. Prentice Hall Inc. 1996,
6. Weste, N.H.E., Eshraghian, K.: “*Principles of CMOS VLSI Design*”. Addison-Wesley, 1994.



# TEMA 12

---

## ***Memorias de Acceso Secuencial***

---

### **+ CONTEXTO**

*Bajo el nombre general de circuito de memoria hemos estudiado primero las EPROM y EEPROM en el tema 7, después las SRAM, las DRAM y las CAM en el tema 11 y ahora vamos a estudiar las memorias FIFO. A lo largo del texto hemos usado esencialmente dos criterios de clasificación: El carácter volátil o permanente de la información almacenada y la forma de acceder a una dirección concreta en la que queremos leer o escribir una palabra de  $n$  bits. Si nos fijamos sólo en el modo de acceso a las distintas direcciones, vemos que hay esencialmente tres modos distintos: (1) el acceso aleatorio, en el que nos cuesta lo mismo acceder a cualquier dirección, (2) el acceso por contenido y (3) el acceso secuencial que vamos a estudiar en este tema.*

*La característica distintiva de las organizaciones de acceso secuencial es que se sustituye toda la electrónica del direccionamiento directo por una organización propia de registros de desplazamiento ayudada en algunos casos por punteros que señalan donde está la primera dirección*

*no ocupada (puntero de escritura) y donde está la última dirección ocupada (puntero de lectura).*

*El soporte físico de estas memorias de acceso secuencial está en las celdas SRAM en CMOS que ya hemos estudiado y en los dispositivos acoplados por carga CCD ( Charge Coupled Devices), que estudiaremos en este tema, con el que terminamos el estudio de los distintos tipos de circuitos de memoria.*

## **++ CONOCIMIENTO PREVIO NECESARIO**

*Todo lo que necesitamos para estudiar este tema ya lo conocemos de temas anteriores, salvo los circuitos CCD y las etapas dinámicas en CMOS a las que dedicaremos sendos apartados en el tema. Las celdas RAM de las FIFO son las mismas que hemos visto en el tema 11.*

## **+++ OBJETIVOS DEL TEMA**

**Objetivo 1:** *Conocer las distintas organizaciones de acceso secuencial (FIFO, LIFO y registros CCD).*

**Objetivo 2:** *Conocer las bases físicas de las etapas dinámicas en MOS, CMOS y CCD.*

**Objetivo 3:** *Comprender las funcionalidades de las memorias FIFO actuales basadas en grandes bloques de memoria RAM y actuando básicamente como buffers de alta velocidad.*

**Objetivo 4:** *Conocer algunos ejemplos de FIFOs asíncronas y síncronas y mostrar algunos de sus aplicaciones.*

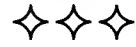
## **++++ GUÍA DE ESTUDIO**

*En este tema seguimos la estrategia usual de comentar primero los aspectos básicos de organización, después estudiar las estructuras electrónicas soporte y, finalmente, ilustrar con circuitos reales el estado actual de la tecnología electrónica en el campo de las memorias FIFO.*

*Creemos que el tema no debe presentar especiales dificultades a estas alturas del texto, en las que ya estamos acostumbrados a analizar cronogramas y describir estructuras internas y significados funcionales de un conjunto de terminales. Quizá las secciones sobre CCD y etapas*



*dinámicas en MOS puedan presentar algo más de complejidad. Por eso se ha desarrollado un conjunto de ejercicios complementarios sobre estos apartados en el texto de problemas.*



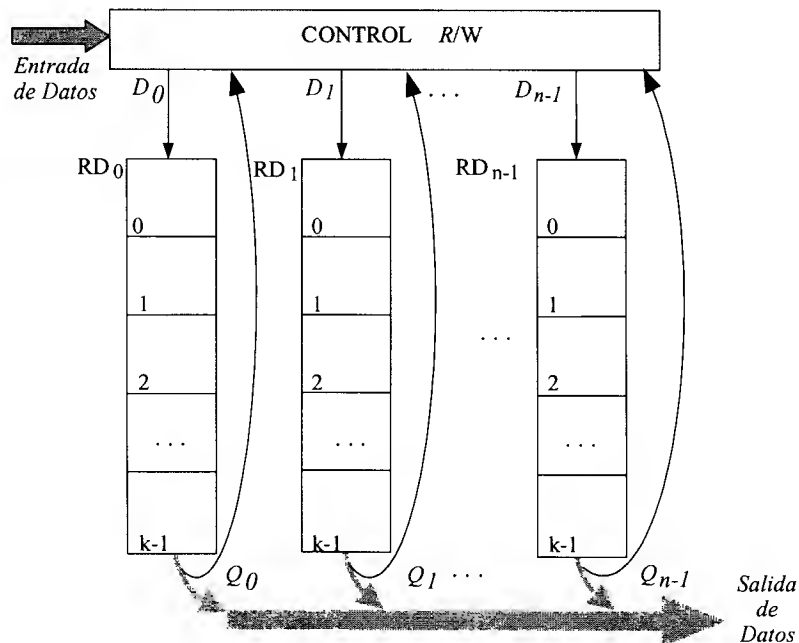
---

## Contenido del Tema

---

### 12.1. ORGANIZACIONES DE ACCESO SECUENCIAL

Veremos primero las organizaciones de memorias de acceso secuencial basadas en registros de desplazamiento. En la *figura 12.1* se muestra la organización FIFO (First-In, First-Out) en la que una memoria de  $K$  palabras de  $n$  bits se construye a partir de  $n$  registros de desplazamiento (uno por bit) de longitud  $K$  (número de palabras).

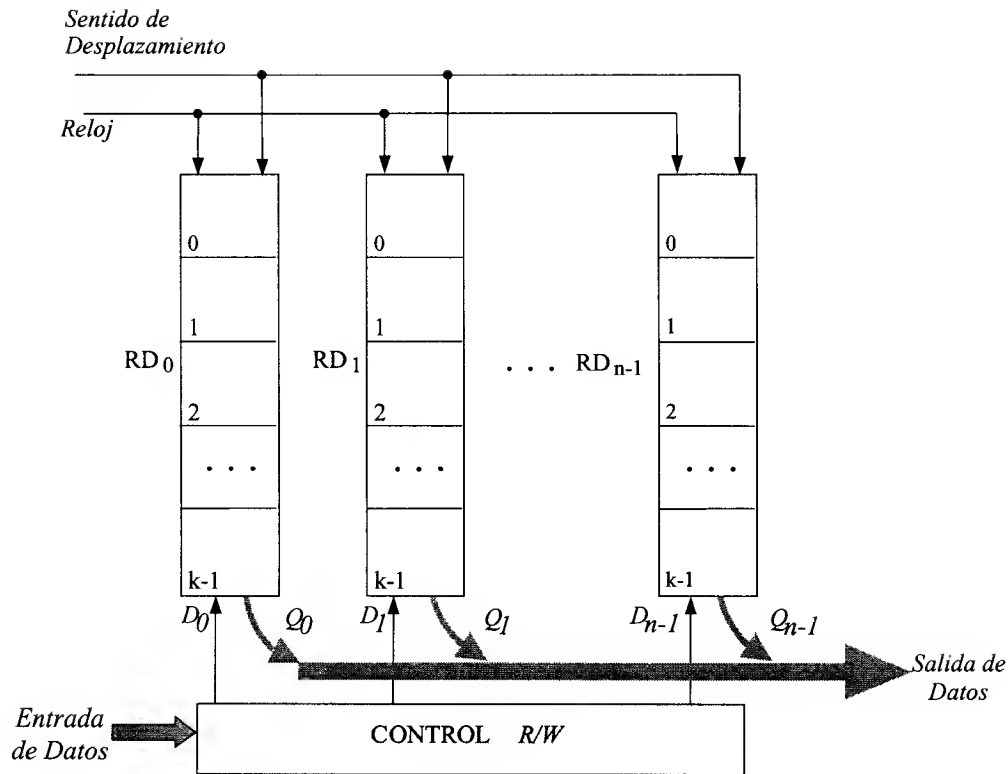


**Figura 12.1.** Organización FIFO de memorias de acceso secuencial. El primer dato que entra es también el primero que aparece accesible a la salida.

Es una organización serie-serie porque los datos se leen en serie y se escriben en serie y es FIFO porque a cada pulso de reloj se desplazan una posición los contenidos de todos los registros de desplazamiento, de forma que la palabra que entró primero, es la que también sale primero. Si no hay entrada de nuevos datos y el lazo está cerrado, la información recircula. Esta constante lectura y reescritura de datos reconstruye la carga perdida en las celdas y permite mantener la información de forma análoga a como ocurría con los mecanismos de refresco para las memorias RAM dinámicas. El control de lectura/escritura selecciona el modo de operación. Cuando se escribe se hace en  $(RD_0, RD_1, \dots, RD_{n-1})$ . Cuando se lee, se toman los datos de  $(Q_0, Q_1, \dots, Q_{n-1})$ . Para cada bit de la palabra, entre  $RD_i$  y  $Q_i$ , existe un registro de desplazamiento formado por  $K$  biestables D.

Una organización diferente es la que se muestra en la *figura 12.2*. Para la misma capacidad y longitud de palabra ( $n$  registros de  $K$  bits), los datos se introducen y se toman del mismo punto, de forma que los sucesivos impulsos del reloj "empujan" hacia arriba las palabras o las "dejan caer", dependiendo del valor del bit que marca el sentido del desplazamiento de los registros que ahora han de ser bidireccionales. Supongamos que estando facilitada la escritura, el bit de sentido de desplazamiento está en alta indicando, por ejemplo, desplazamientos "hacia arriba". Cada pulso de reloj entra una nueva palabra que empuja a las que habían entrado antes aumentando la altura de la "pila". Si cambiamos ahora de lectura a escritura hay que cambiar también el sentido del desplazamiento y aparecen en el primer registro las palabras en orden inverso al que fueron escritas. Esta organización se llama LIFO (Last-In, First-Out) ya que la última palabra que entró en la memoria es la primera que sale cuando el control pasa de escritura a lectura. También se le suele llamar memorias de *pila* ("stack") porque las palabras se "amontonan" en forma de pila.

Cuando queremos aumentar la capacidad de las memorias de acceso secuencial buscamos celdas básicas dinámicas y sencillas para que ocupen poca superficie en el semiconductor y métodos de direccionamiento que sean también sencillos. En cuanto a las celdas básicas las opciones dominantes son las celdas RAM que ya hemos visto en el tema anterior y las estructuras CCD que veremos más adelante en este tema. Ahora bien, al aumentar mucho la longitud de los registros de desplazamiento el tiempo de acceso a la información crece de forma tal que las organizaciones serie-serie (tipo LIFO ó FIFO) no son adecuadas. Siempre hay que moverse en una situación de compromiso entre capacidad y tiempo de acceso.

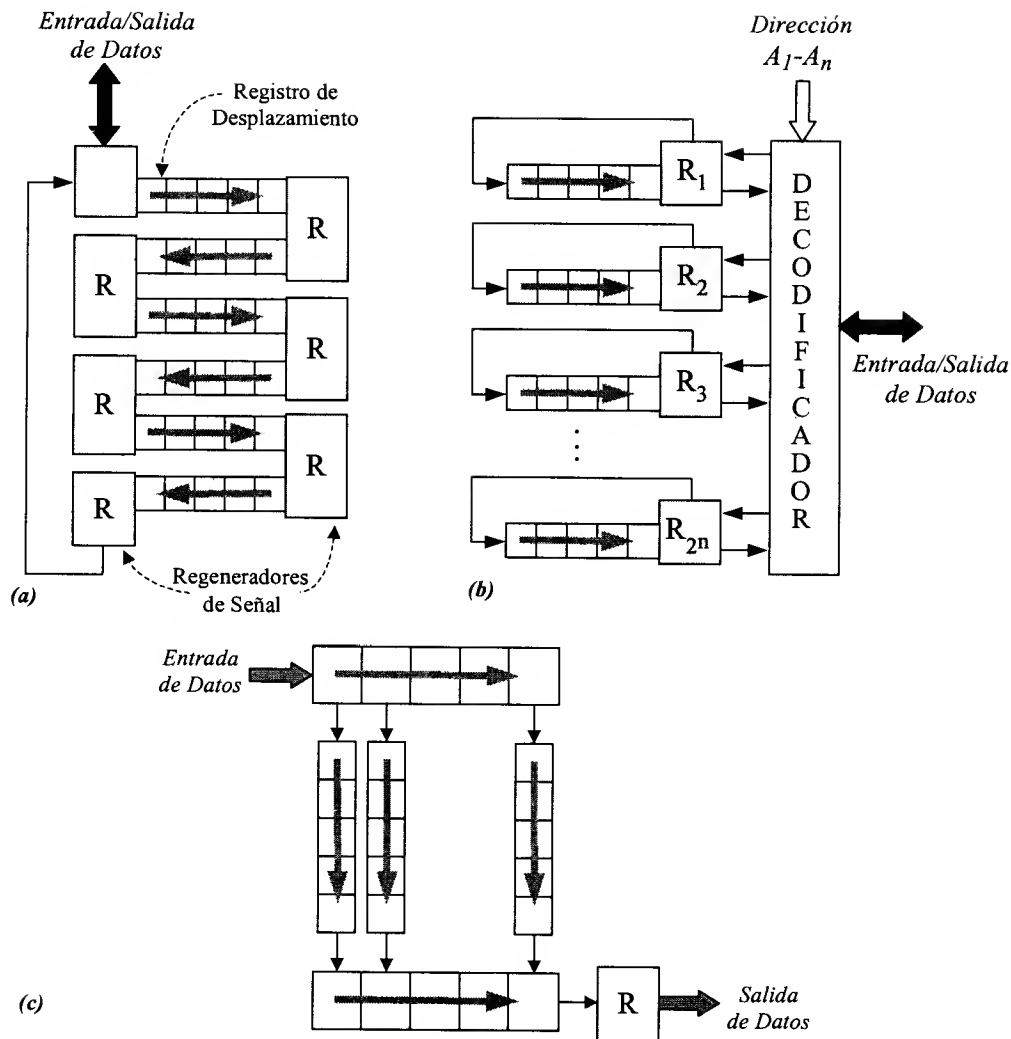


**Figura.12.2.** Organización LIFO de memorias de acceso secuencial. El primer dato que sale es el último que entró.

La figura 12.3 muestra otras organizaciones propias de las memorias CCD. En 12.3.a se mantiene la organización serie incluyendo etapas regeneradoras de señal en todas las “esquinas”. En esta organización todos los bits realizan el mismo recorrido a través del lazo y a la misma frecuencia. El número de bits entre dos amplificadores-regeneradores sucesivos está determinado o bien por la eficiencia en el proceso de transmisión (inversamente proporcional a las pérdidas de carga) o bien por el valor más bajo en la frecuencia de trabajo deseada en condiciones de mantenimiento. Evidentemente, como esta organización es serie, el número de bits máximo (longitud del ciclo de recirculación) determina el valor medio del tiempo de acceso.

La figura 12.3.b introduce una pequeña modificación en la organización anterior que mejora el tiempo de acceso al coste de aumentar la electrónica de direccionamiento. Lo que se ha hecho es segmentar en tramos y direccionar cada uno de estos tramos de forma directa, como hacíamos en las

memorias RAM.



**Figura 12.3.** Organización de memorias CCD. **(a)** Organización serie, con etapas intermedias de regeneración. **(b)** Organización por lazos múltiples con direccionamiento directo para cada lazo. **(c)** Organización S-P-S (serie-paralelo-serie).

A cada segmento se le llama lazo. Dentro del lazo hay dos registros CCD y dos amplificadores regeneradores y cada lazo es considerado como una celda de memoria RAM direccionable a través de

un multiplexo que decodifica los bits de dirección y una lógica de control que facilita el acceso al lazo seleccionado para realizar allí las operaciones de lectura y escritura. Ahora el máximo tiempo de acceso para un bit arbitrario es  $t_A = 2(N_r / f_c)$  siendo  $N_r$  el número de bits en cada segmento, entre los amplificadores de regeneración, y  $f_c$  la frecuencia del reloj.

La tercera organización es la que se muestra en la *figura 12.3.c*, llamada serie-paralelo-serie (SPS). Consta de dos registros serie y un gran registro paralelo multicanal. Los datos se introducen en serie en el registro superior. Después se transfieren en paralelo a la primera etapa del registro paralelo. Todos los canales paralelos se desplazan al unísono con un mismo reloj más lento que el de los registros serie y a la salida se realiza el proceso inverso. El registro más bajo es de nuevo serie y rápido, se carga en paralelo y se lee a través del amplificador regenerador.

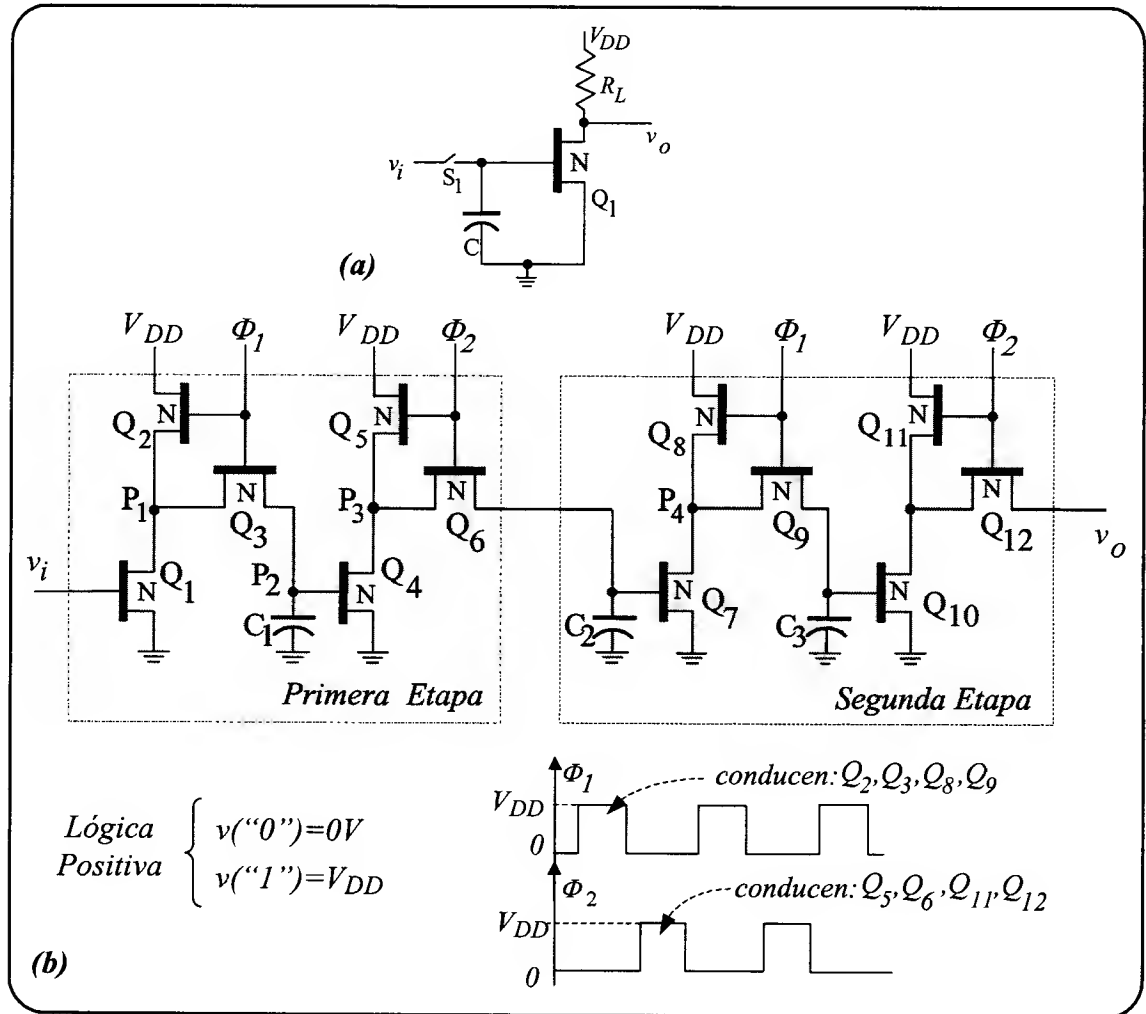
Obsérvese que en esta organización SPS todos los bits no siguen el mismo camino. Si el registro serie tiene  $N_s$  bits y el paralelo,  $N_p$ , cada bit se transfiere a través de  $N_s + N_p$  etapas. Así, basta con un amplificador regenerador para mantener la información asociada a un número de bits mucho mayor que si la organización fuera serie.

## 12.2. ETAPAS DINÁMICAS EN MOS Y CMOS

Para realizar las etapas de almacenamiento transitorio junto con el control local de la transferencia entre etapas vecinas puede usarse celdas SRAM convencionales y lógica combinacional. Sin embargo, para el tamaño usual en memorias de acceso secuencial y en particular para las CCD, estos diseños ocupan mucha área de semiconductor por lo que se usan etapas dinámicas en MOS y CMOS.

Debido a la alta impedancia de entrada de los transistores MOS existe la posibilidad de almacenar carga en la capacidad de puerta. El circuito de la *figura 12.4.a* muestra la versión en tecnología discreta de esta configuración que consta simplemente de un condensador,  $C$ , en paralelo con los terminales de puerta y fuente del transistor  $Q_1$  y conectado con la línea de datos,  $v_i$ , a través de un conmutador,  $S_1$ . Al aplicar una tensión de entrada,  $v_i$ , en la puerta y en el condensador a través del conmutador  $S_1$ , el condensador se carga. Al abrir  $S_1$ , el condensador retiene la carga durante bastante tiempo puesto que la única vía de descarga es a través de la impedancia de entrada del transistor, reteniéndose la información almacenada y actuando, por consiguiente, como una celda de memoria volátil. La salida de esta celda de memoria se toma del drenador de  $Q_1$ , de forma que el condensador

nunca se descarga al leerlo. En la práctica la carga en el condensador termina por perderse tal como hemos comentado antes, puesto que el conmutador  $S_1$  se realiza físicamente con otro transistor MOS que ofrece cierta resistencia finita. Esta resistencia junto con la de entrada de  $Q_1$  termina por descargar el condensador.

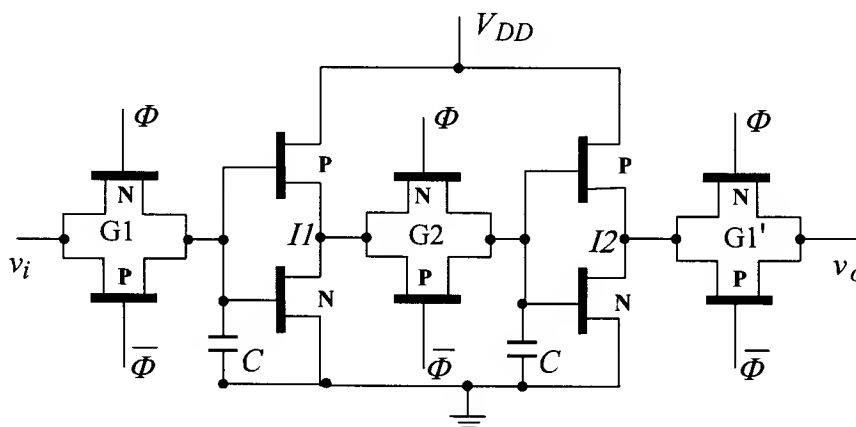


**Figura 12.4** Circuitos MOS para registros de desplazamiento en lógica dinámica. (a) Estructura básica de la celda. (b) Ejemplo de registro de desplazamiento usando reloj bifásico.

La *figura 12.4.b* muestra el esquema de uno de los primeros registros de desplazamiento en lógica dinámica (desarrollado por Philco en 1965) que utiliza celdas de este tipo. Para su funcionamiento necesita un reloj de dos fases ( $\Phi_1$ ,  $\Phi_2$ ).

Supongamos que se introduce un “0” en la entrada. Durante el intervalo en el que el reloj  $\Phi_1$  está en alta los transistores  $Q_2$  y  $Q_3$  pasan a conducción. Consecuencia del “0” en la entrada  $Q_1$  está en corte y el punto  $P_1$  es llevado a las proximidades de  $V_{DD}$  por el transistor de carga  $Q_2$ . Este nivel de tensión alto (“1”) se transfiere a través de  $Q_3$  al condensador  $C_1$ . Comienza entonces la fase de reloj  $\Phi_2$ , pasando este a alta y haciendo que ahora conduzca  $Q_5$  y  $Q_6$ . Como consecuencia del “1” almacenado en  $C_1$ , el transistor  $Q_4$  también pasa a conducir y lleva al punto  $P_3$  a tierra. Esta información se transmite a  $C_2$  a través de  $Q_6$  y queda almacenada en  $C_2$  cuando termina el ciclo de reloj  $\Phi_2$ , que vuelve a pasar a corte a los transistores  $Q_5$  y  $Q_6$ . Así, tras  $\Phi_1$  y  $\Phi_2$  se ha desplazado el “0” desde la entrada de la primera etapa,  $v_i$ , a la entrada de la segunda (terminal de puerta de  $Q_7$ ). De forma análoga durante los siguientes ciclos de  $\Phi_1$  y  $\Phi_2$  esta operación se repite también en la segunda etapa a la vez que en la primera etapa se almacena la nueva entrada teniendo, por consiguiente, la estructura básica de un registro de desplazamiento.

Una versión análoga a esta etapa de registro usando tecnología CMOS es la que se muestra en la *figura 12.5*.



**Figura 12.5.** Etapa de registro en CMOS

Para comprender la estructura general del diseño de una etapa de registro de desplazamiento conviene recordar que el diseño se basa en tres puntos: 1) Usar las capacidades parásitas para



almacenar de forma transitoria paquetes de carga. (2) Usar inversores para controlar los estados. (3) Usar puertas de transmisión para facilitar la carga y descarga de esas capacidades bajo el control de un reloj monofásico o polifásico. Pues bien, el circuito de la *figura 12.4* usaba transistores NMOS y un reloj bifásico, en cambio el circuito de la *figura 12.5* usa un reloj monofásico y su complementario y sustituye los inversores NMOS por puertas de transmisión e inversores CMOS.

Recordando el funcionamiento básico del inversor CMOS y de las puertas de transmisión CMOS descritos en el tema 4, no es difícil de comprender el funcionamiento de la etapa de un registro de desplazamiento en CMOS que se muestra en la *figura 12.5*.

En este circuito, la puerta de transmisión facilita el proceso de carga del condensador y contribuye a la descarga con los impulsores de los inversores CMOS. Estos inversores mantienen el estado ó conmutan, empujando hacia delante el paquete de carga que almacena el condensador, al ritmo del reloj monofásico,  $\Phi$ .

### 12.3. ESTRUCTURAS CCD

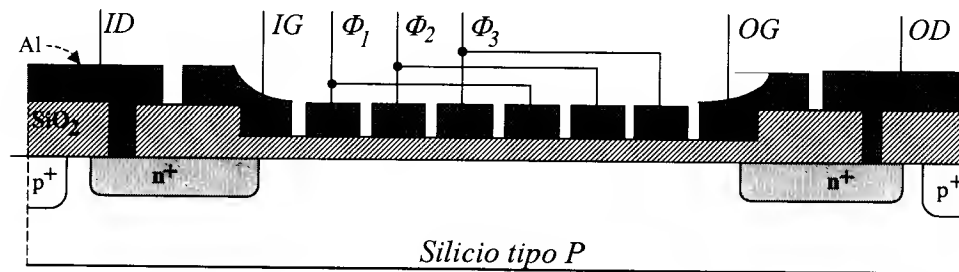
Un dispositivo CCD es básicamente una sucesión de estructuras MOS que pueden almacenar paquetes de carga en pozos de potencial. Estos paquetes se pueden *inyectar*, *transferir* y *extraer* bajo el control de un reloj polifásico que desplaza el pozo de potencial a lo largo de la superficie (SCCD) o en un canal enterrado (BCCD). Así, un CCD actúa como un registro de desplazamiento en el que la información no está representada por tensiones o corrientes sino por paquetes de carga. Desde su invención por Boyle-Smith y Kosonocky-Saver en 1970, su desarrollo ha sido muy amplio en aplicaciones tales como:

- Memorias digitales.
- Formación de imágenes (sensor óptico).
- Procesamiento de señales analógicas.

Esto ha sido posible porque de hecho CCD no es una nueva estructura sino una nueva forma de usar las conocidas estructuras MOS en tecnología de silicio, perfectamente conocida también. De hecho los dispositivos CCD pueden considerarse como transistores MOS multipuerta.

La *figura 12.6* muestra un corte transversal de un dispositivo CCD superficial de 2 bits mandado por un reloj de 3 fases que actúa como un registro de desplazamiento con 3 secciones:

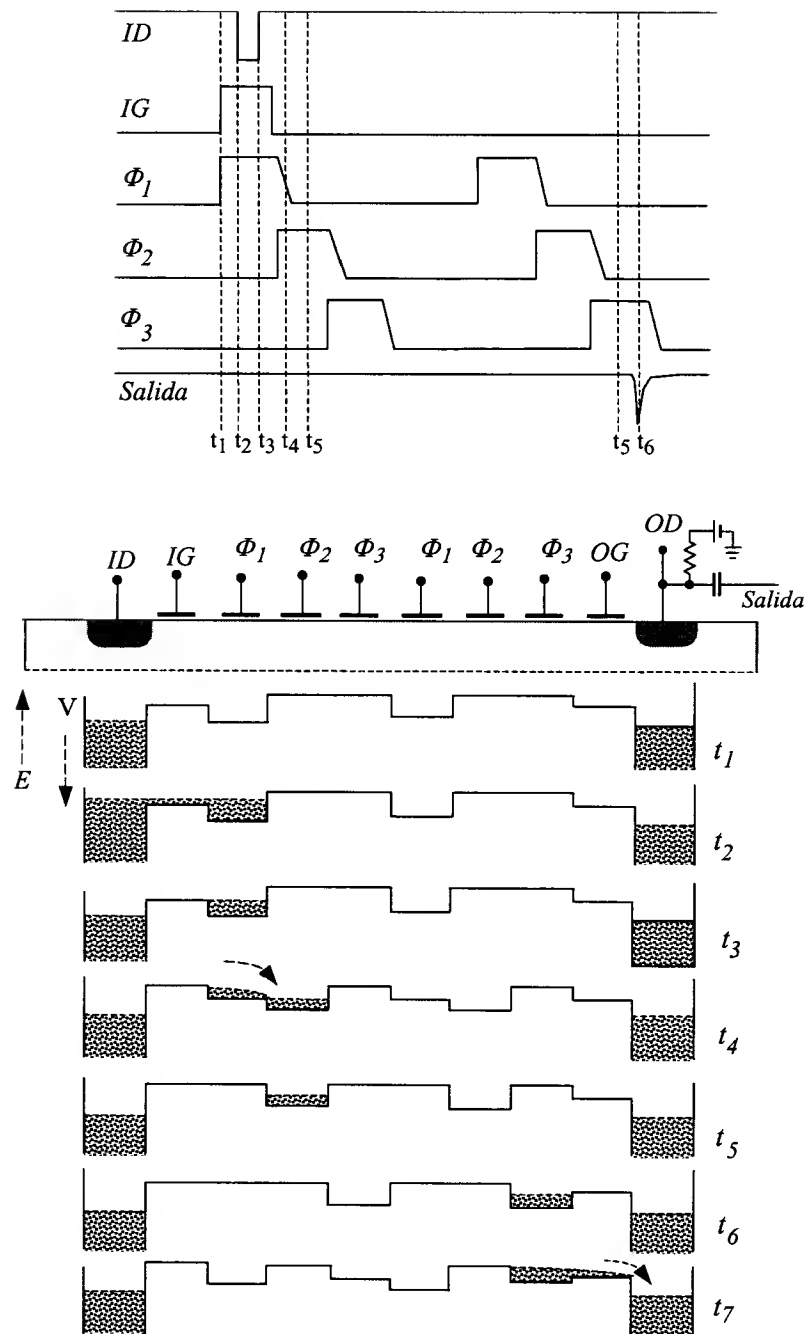
1. *Sección de entrada.* Con un diodo y una puerta que inyectan paquetes de carga en la estructura (ID) bajo control de un potencial externo y una puerta (IG) con un control todo-nada del paso de la carga desde el diodo de inyección al primer pozo de potencial. La difusión p<sup>+</sup> que rodea el área activa evita la inversión del sustrato (P).
2. *Sección de transferencia.* Consiste en una sucesión ordenada de contactos con las 3 fases del reloj ( $\Phi_1, \Phi_2, \Phi_3; \Phi_1, \Phi_2, \Phi_3; \Phi_1, \dots$ ), tantas como bits, que controlan la forma y posición del pozo de potencial en la interface Si-SiO<sub>2</sub> y arrastran la carga desde la entrada a la salida.
3. *Sección de salida.* Consta básicamente de una capacidad construida a partir de una unión P-N polarizada en sentido inverso, cuya tensión cambia cuando se le transfiere un paquete de carga detectando así la llegada del paquete. El diodo se “pone a cero” (reset) mediante un conmutador de “reset”, quedando preparado para la recepción de un nuevo paquete de carga. Este nudo suele estar conectado a un transistor MOS, como veremos más adelante.



**Figura 12.6.** Corte transversal de un dispositivo CCD superficial de 2 bits mandado por un reloj de 3 fases, ( $\Phi_1$ ,  $\Phi_2$  y  $\Phi_3$ ) que actúa como un registro de desplazamiento.

La carga puede introducirse de forma serie a través de la difusión de entrada o por absorción de fotones en las proximidades del pozo de potencial. Supongamos sin embargo que la inyección es por ID y veamos como se traslada la carga a lo largo de la estructura (figura 12.7).

En  $t=t_1$ ,  $\Phi_1$  está en alta y  $\Phi_2$  y  $\Phi_3$  están en baja. Los diodos de entrada y salida (ID, OD) están polarizados en alta para evitar situaciones de inversión. Así, las superficies bajo IG y OG están en situación de fuerte vaciamiento y ID y OD no pueden suministrar carga a la estructura CCD. Es decir, todos los pozos de potencial están vacíos. El pozo de potencial bajo  $\Phi_1$  en  $t=t_1$  es más profundo que bajo  $\Phi_2$  y  $\Phi_3$ , porque en este intervalo de tiempo es mayor el potencial aplicado a  $\Phi_1$ .



**Figura 12.7.** Análisis gráfico del proceso de transporte controlado de la carga mediante pozos de potencial.

En  $t=t_2$ , se baja el potencial de  $ID$  y se inyectan electrones en el pozo que hay bajo  $\Phi_1$ . Al final de la inyección, los potenciales bajo  $IG$  y  $\Phi_1$  son los mismos que bajo el diodo de entrada. Así quedan almacenados los electrones bajo  $\Phi_1$  y  $IG$ .

En  $t=t_3$ , el potencial de  $ID$  vuelve a alta, los electrones bajo  $IG$  y el exceso de electrones bajo el primer electrodo  $\Phi_1$  salen fuera del dispositivo por el terminal  $ID$ , creando un paquete de carga de valor bien definido bajo  $\Phi_1$ .

En  $t=t_4$  el potencial de  $\Phi_1$  vuelve a bajar mientras que  $\Phi_2$  está en alta. Este proceso se llama transferencia de carga. El potencial bajo  $\Phi_1$  baja en pendiente lentamente, porque las cargas necesitan un tiempo finito para pasar por debajo del electrodo.

En  $t=t_5$ , se completa el proceso de transferencia de carga y el paquete queda almacenado bajo  $\Phi_2$ . Este proceso se repite en  $t=t_6$  (el paquete pasa bajo el segundo electrodo  $\Phi_3$ ).

En  $t=t_7$ , el potencial de los electrodos  $\Phi_3$  vuelve a baja y empuja a los electrones hacia el diodo de salida, generando un potencial en la salida proporcional al valor de la carga transferida.

Puesto que los paquetes de carga siguen la posición espacial del pozo de potencial y estos se pueden controlar por tensiones externas, la transferencia de carga se puede realizar en las dos direcciones del plano.

Por otro lado, las señales inyectadas pueden ser tanto analógicas como digitales y el dispositivo puede tener entradas y salidas múltiples. Su consumo es bajo y la densidad de integración muy alta.

Aunque inicialmente, las aplicaciones de las estructuras CCD cubrían las tres áreas que hemos mencionado al comienzo (memorias de acceso secuencial tipo FIFO, procesadores de señales analógicas y sensores de imágenes), en la actualidad su principal aplicación está en los sensores de imágenes, en las video cámaras. En cambio, cuando la aplicación tiene que ver con la función de buffer entre dos sistemas digitales que operan a distinta velocidad, la tecnología dominante es la CMOS que hemos visto en el tema anterior.

## **12.4. MEMORIAS FIFO SOBRE CELDAS RAM EN CMOS**

La función principal de las memorias FIFO es el almacenamiento transitorio de datos en aquellas situaciones de diseño electrónico en las que sea necesario acoplar dos sistemas digitales que

operen a distinta velocidad y necesiten intercambiar datos. Siempre que los datos lleguen a un procesador en paquetes, de forma irregular o de forma regular pero a mayor velocidad, hace falta un almacenamiento intermedio, un buffer, en el que los datos que se escriben primero son los que primero salen.

Tenemos muchos símiles de la necesidad de esta función. Por ejemplo las funciones de un depósito regulador de caudal en el que el agua sale y debemos conseguir que el depósito no rebose ni se quede vacío. Las colas que se forman en la caja de un supermercado o las aglomeraciones de coches en un semáforo cerrado son otros ejemplos. Las características que definen la necesidad de las memorias FIFO son:

- 1) Acceso a sistemas de proceso lento pero constante con datos (demandas de servicio) que llegan de forma irregular.
- 2) Acceso a sistemas que procesan en paquetes pero a los que los datos llegan de forma esporádica.
- 3) Interfaces entre sistemas que trabajan a distinta velocidad.

Estas necesidades que aparecen en procesadores de señales y en comunicación con periféricos, por ejemplo pueden satisfacerse a nivel electrónico, con un circuito de propósito específico, o por programación. La primera solución es más rápida (hasta 3,6 Gbits por segundo), la segunda más adaptable ante cambios de requerimientos.

Los datos pueden escribirse en una FIFO a una velocidad y leerse a otra diferente y el tamaño de la FIFO necesaria en una aplicación de comunicación entre CPUs o entre una CPU y un periférico, depende del tamaño de los paquetes de datos a transmitir y de la diferencia entre las velocidades de los circuitos procesadores que conecta. Cuanto mayor es esta diferencia, más datos deberá poder almacenar para que no se quede vacía ni fuerce ciclos de espera en el procesador. La situación usual es que el procesador periférico sea mucho más lento que el procesador central.

El tiempo de ciclo ( $t_c$ ) de una FIFO determina la velocidad del sistema en el que puede operar esta memoria sin necesidad de requerir estados de espera en el procesador. El tiempo de ciclo posee dos componentes: tiempo de acceso ( $t_A$ ) y tiempo de recuperación ( $t_R$ ), de forma que,  $t_c = t_A + t_R$ .

Obviamente, la máxima frecuencia de operación es  $f_{max} = 1/t_c$ . El tiempo de acceso es el tiempo asignado para completar la operación en curso y el de recuperación el necesario para que la FIFO

vuelva al estado estacionario y quede preparada para ejecutar la siguiente operación.

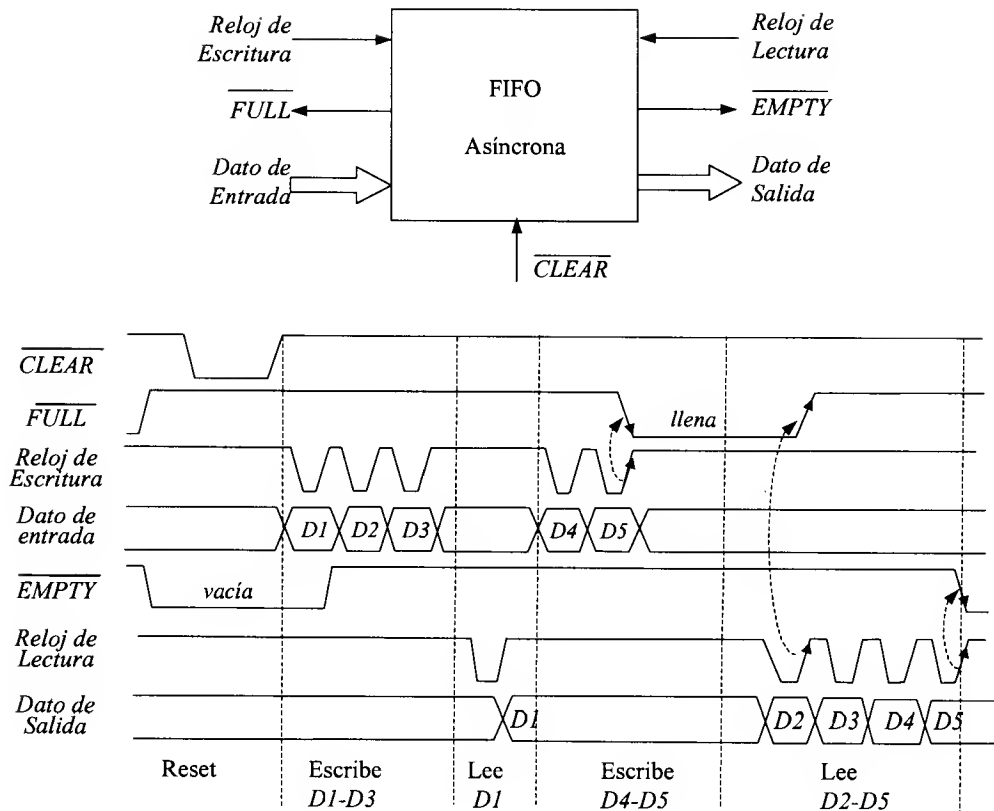
### 12.4.1. Tipos de FIFO

Hay básicamente tres tipos de memorias FIFO.

- I:** FIFO *tipo registro de desplazamiento*, que ya hemos visto en el apartado anterior. En estas el número de palabras almacenadas es fijo (coincide con la longitud del registro) y hay un sincronismo implícito y necesario entre las operaciones de lectura y escritura. A medida que van entrando nuevas palabras dato en los registros FIFO, otras van saliendo por el otro extremo.
- II:** FIFO *de lectura/escritura mutuamente exclusivas* en las que, en cada momento, sólo se puede leer o escribir, pero no ambas cosas. El número de palabras almacenadas es variable y deben satisfacerse ciertas condiciones en el cronograma entre las señales procedentes del “sistema que escribe” y las procedentes del “sistema que lee”. Es necesario un cierto nivel de sincronismo entre estos dos sistemas.
- III:** FIFO *de lectura/escritura concurrente*, con un número variable de palabras almacenadas y posibilidad de lectura y escritura asíncrona, pudiendo coexistir ambos procesos. Es decir, no hay restricciones en el cronograma de los ciclos de lectura y escritura. Son independientes y no necesitan ningún sincronismo entre ellos. Esto significa que cuando dos sistemas de distinta frecuencia se conectan a la FIFO, no necesitamos preocuparnos de la sincronización, porque la realiza internamente el circuito.

Todas las soluciones actuales sobre memoria RAM en CMOS y orientadas a funciones de “buffer” entre dos sistemas electrónicos de distinta velocidad son de los tipos II ó III, es decir de longitud de palabra variable. Los primeros diseños fueron del tipo II pero actualmente, prácticamente todas las FIFO son del tipo III, con posibilidad de lectura y escritura independientes y con almacenamiento de un número de palabras variable. Esto les permite integrarse en cualquier sistema digital síncrono bajo el control de un único reloj central.

A su vez las FIFO concurrentes pueden ser *asíncronas* o *síncronas*, dependiendo del tipo de señales de control usadas en los procesos de lectura y escritura. La *figura 12.8* muestra los terminales de una FIFO asíncrona y la *12.8.b*, el cronograma para una profundidad de cuatro palabras y dos ciclos, uno de escritura de las palabras  $D_1$ ,  $D_2$  y  $D_3$ , seguido de la lectura de  $D_1$  y seguido de nuevo de la escritura de  $D_4$ ,  $D_5$  y, finalmente, la lectura de  $D_2$ ,  $D_3$ ,  $D_4$  y  $D_5$ .



**Figurta 12.8. (a)** Terminales de una FIFO asíncrona. **(b)** Cronograma para longitud 4.

Este tipo de circuitos de memoria posee buses para entrada y salida de datos, relojes separados para el control de los ciclos de lectura y escritura, indicadores del estado de ocupación de la memoria ( $\overline{FULL}$ ,  $\overline{EMPTY}$ ) y señal de reset ( $\overline{CLEAR}$ ).

Las líneas *reloj de escritura* y  $\overline{FULL}$  se usan para escribir datos. Antes de escribir una palabra en una FIFO es necesario comprobar si hay o no espacio disponible, lo que se sabe a través de la línea  $\overline{FULL}$ . Si hay espacio libre, la palabra se escribe a la bajada del pulso del reloj de escritura.

De forma análoga, las líneas de *reloj de escritura* y  $\overline{EMPTY}$  se usan para leer. Si no hay datos, no tiene sentido leerlos. Si los hay se leen a la caída del pulso del reloj de lectura.

Al comienzo de un ciclo de lectura o escritura es necesario establecer condiciones iniciales de

reset. En el ejemplo de la figura, tras el pulso de  $\overline{CLEAR}$ , se escriben tres palabras ( $D_1$ ,  $D_2$  y  $D_3$ ) en tres pulsos sucesivos del reloj de escritura. Una vez que se ha introducido la primera palabra, la señal  $\overline{EMPTY}$  cambia de baja a alta. Después se escriben otras dos palabras, antes de que pase a baja el reloj de lectura, que inicia un ciclo de lectura. La lectura del primer dato no cambia el valor de las señales de status ( $\overline{FULL}$ ,  $\overline{EMPTY}$ ), porque ni está llena del todo, ni vacía, con las operaciones que hemos realizado.

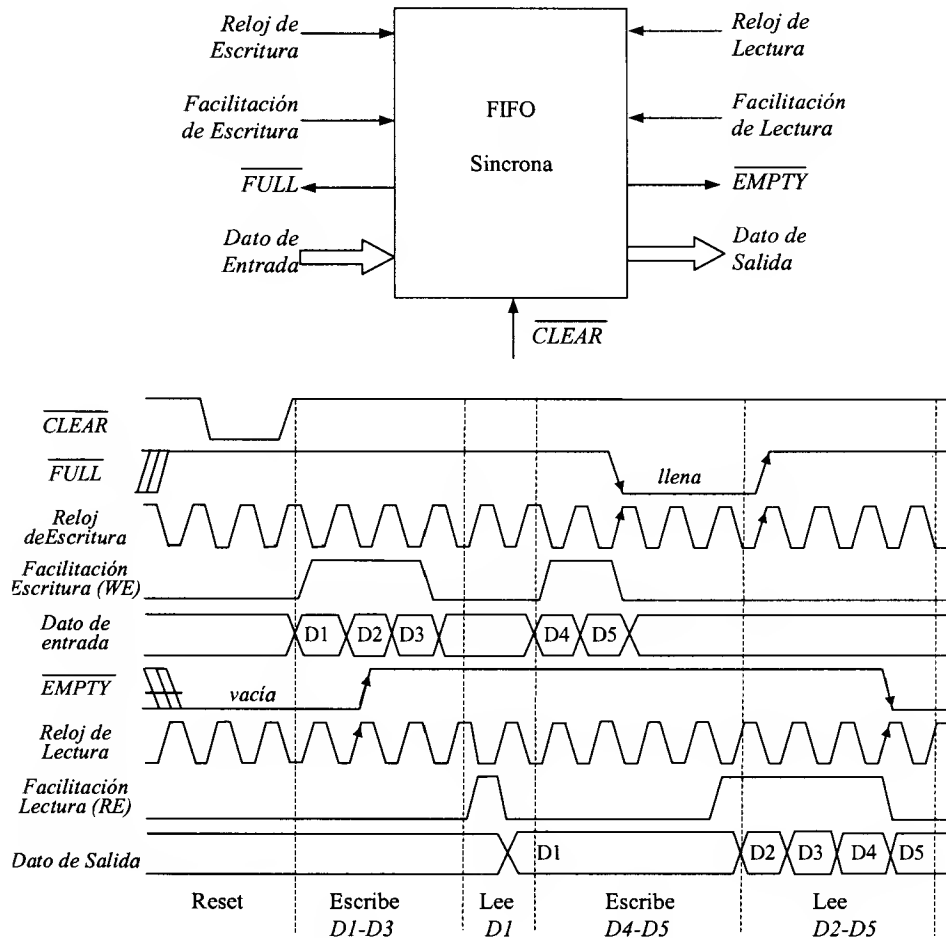
Si ahora escribimos dos nuevas palabras, la FIFO estará llena, tal como indica el cambio de estado en  $\overline{FULL}$ . Finalmente, las cuatro palabras que quedan todavía en la FIFO, ( $D_2 \dots D_5$ ), se leen, vaciando de nuevo la memoria, tal como indica la señal  $\overline{EMPTY}$ , que pasa entonces a baja. La desventaja de una FIFO de esta clase es que las señales de “*status*” no pueden ser sincronizadas totalmente con los relojes de lectura y escritura.

Si hay espacio en la FIFO para sólo una palabra dato, el siguiente ciclo de escritura activa a  $\overline{FULL}$ . Entonces, el sistema de escritura “interroga” a la señal  $\overline{FULL}$  y espera hasta que, de nuevo, haya espacio libre en la FIFO. En el momento en que se lea una palabra, el reloj de lectura pone a cero la línea status  $\overline{FULL}$ . Ahora bien, este reset, que es *síncrono* con el sistema de lectura, es *asíncrono* con el sistema de escritura y en el peor de los casos no deja tiempo para que transcurran los retardos necesarios para el asentamiento de las señales dando lugar a potenciales problemas de inestabilidad. Este problema puede presentarse también con la señal  $\overline{EMPTY}$  que debe de ser síncrona con el sistema que lee la FIFO pero que quien la activa es el sistema que escribe.

Para resolver estos problemas se usan FIFOs síncronas, que actúan bajo el control de un único reloj central, como cualquier otro componente de un sistema digital síncrono. Este reloj está siempre activo y subyace a la operación de todos los circuitos, estén o no ejecutando acciones en un momento determinado. Las señales de facilitación (*chip select*) inician la ejecución síncrona de las operaciones de lectura y escritura en todos los circuitos del sistema. En la *figura 12.9.a* mostramos los distintos terminales de una FIFO síncrona, donde junto a las líneas que ya conocemos, aparecen dos señales de reloj, una procedente del sistema que escribe y otra del que lee. En el cronograma de la parte *b* de la *figura 12.9* muestra como se inicia la lectura con el flanco de bajada de  $WE$  y los datos se escriben con el flanco de subida del siguiente pulso de reloj. Análogamente  $RE$  controla la lectura. Obsérvese que  $WE$  está sincronizada con el reloj de escritura y  $RE$  con el de lectura. También hay sincronismo de



$\overline{FULL}$  y  $\overline{EMPTY}$  con los relojes de escritura y lectura. Así, la línea de  $\overline{FULL}$  sólo cambia de nivel en sincronismo con el reloj de escritura, aunque el cambio lo haya provocado la lectura de una palabra.



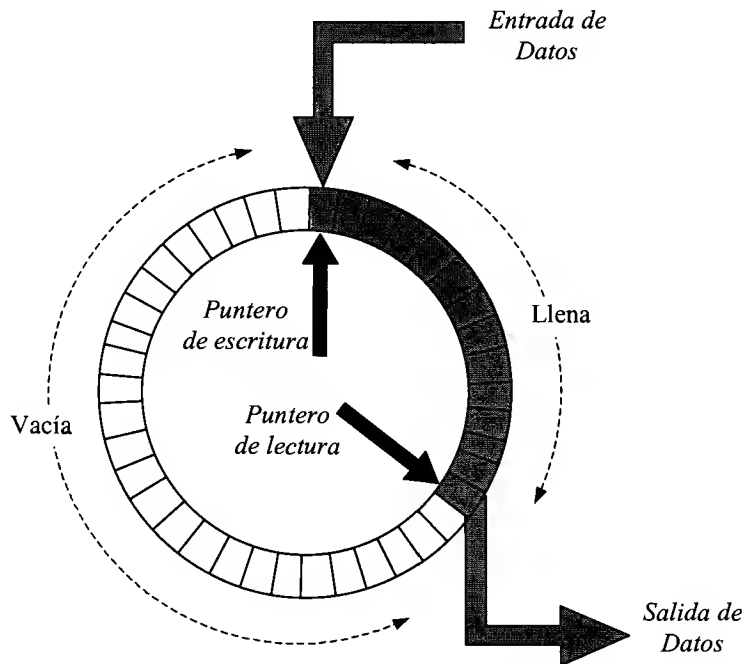
**Figura 12.9.** (a) Terminales de una FIFO síncrona de Texas Instrument. (b) Cronograma de los procesos de lectura y escritura.

### 12.4.2. Arquitecturas de las FIFO-RAM

Hay dos tipos de arquitecturas básicas. La primera es la inherente a los registros de desplazamiento: entra un nuevo dato, que “cae” hasta la primera posición no ocupada, se desplazan

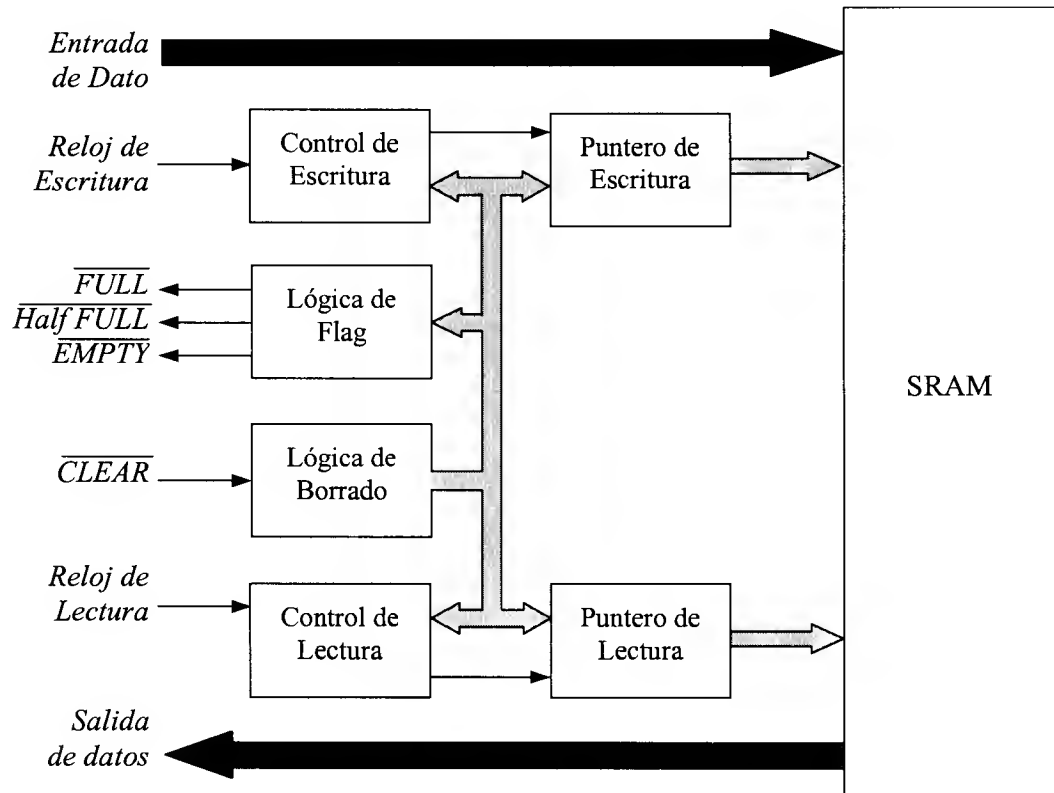
todos los demás y sale el del último biestable del registro. El principal inconveniente de esta organización es el retardo intrínseco al recorrido de todo el registro cuando el tamaño de la FIFO es grande.

El otro tipo de arquitectura está basado en una *organización circular* de la memoria y el uso de dos punteros. Así, nos evitamos tener que desplazar los datos a través de todas las posiciones de la FIFO. La idea de esta arquitectura se ilustra en la *figura 12.10*. Aquí, la dirección de memoria a la que va el dato de entrada es la que señala el puntero de escritura. Análogamente, la primera dirección de la FIFO en la que debe leerse es la que señala el puntero de lectura. Al inicializarse la FIFO tras el reset, los dos punteros indican la misma dirección de memoria. Tras cada operación de escritura, el puntero correspondiente se desplaza a la siguiente posición. Análogamente, cada operación de lectura desplaza el puntero de lectura hasta la dirección de la siguiente palabra que tiene que leerse. Así, el puntero de lectura sigue constantemente al de escritura y cuando lo alcanza es que la FIFO está vacía. Análogamente, si el puntero de escritura alcanza al de lectura, la FIFO está llena.



**Figura 12.10.** Organización circular de los circuitos FIFO con dos punteros..

Los circuitos FIFO que responden a este tipo de organización circular con dos punteros se diseñan a partir de memorias SRAM con entrada y salida separadas y contadores para implementar los punteros que generan en cada momento las direcciones de lectura o escritura, tal como se ilustra en la figura 12.11. Junto a las señales de descripción de los estados de FIFO llena ( $\overline{FULL}$ ) o vacía ( $\overline{EMPTY}$ ), algunas FIFO proporcionan también señales que describen estados intermedios como “semillena” (half FULL), “casi llena” (almost FULL) y “casi vacía” (almost EMPTY).



**Figura 12.11.** Diagrama de bloques de las FIFO SRAM de T. I. tipo ACT-7881.

La principal ventaja de esta arquitectura es que puede aumentarse significativamente la capacidad de la FIFO sin incrementar el tiempo de acceso ni la complejidad de la electrónica de control, salvo al contador binario que, lógicamente, necesita ser de  $n$  bits para  $2^n$  posiciones.

## 12.5. EJEMPLO

Para ilustrar el estado de la tecnología electrónica en circuitos FIFO basados en memoria RAM estática vamos a comentar la FIFO Am7205A de AMD (8192x9). Este circuito está organizado en 8192 palabras de 9 bits, con posibilidades de lectura y escritura asíncrona y simultánea y facilidades de conexión serie y/o paralelo para aumentar la longitud de palabra o el número de palabras.

La figura 12.12 muestra el diagrama funcional de la FIFO. Los terminales  $D0-D8$  son los datos de entrada y los  $Q0-Q8$  los datos de salida. Estos terminales están en el estado de alta impedancia siempre que la línea de control de lectura ( $\overline{R}$ ) está en alta.

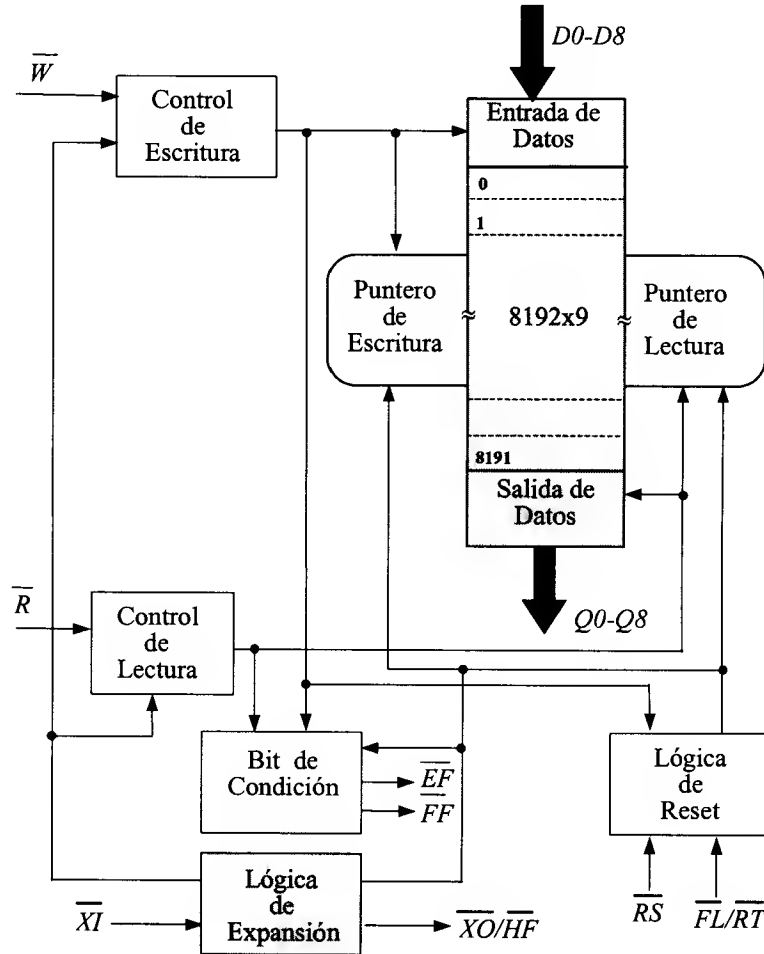
Tal como hemos descrito en el apartado anterior el direccionamiento se realiza por punteros uno para lectura y otro para escritura ya que la FIFO está organizada como una memoria RAM de doble puerta. Cuando estos punteros alcanzan el valor de la dirección más alta (8191), pasan a cero, de forma que los bits que representan el estado lógico de la FIFO siempre operan sobre la diferencia entre los valores de estos dos punteros.

Alrededor de la memoria existe una lógica de control que guía los procesos de lectura y escritura evitando situaciones no convenientes, tales como leer cuando una celda está vacía o escribir sobre una celda llena. Existe también una lógica de control adicional para facilitar la expansión. Veamos primero una breve descripción del significado funcional de los terminales de control, después describiremos los modos básicos de operación (reset, lectura, escritura y retransmisión).

Hay dos bits de condición ( $\overline{EF}$  y  $\overline{FF}$ ), ambos activos en baja.  $\overline{EF}$  (*empty flag*) indica que la FIFO contiene datos y pueden por consiguiente ser leídos.  $\overline{EF}$  pasa a baja cuando el puntero de la dirección lectura es igual al puntero de la dirección escritura, indicando que el dispositivo está vacío. Por consiguiente, mientras  $\overline{EF}$  esté en baja inhibirá posteriores operaciones de lectura. El paso a alta de  $\overline{EF}$  se realiza durante el primer ciclo de escritura de una FIFO vacía, tras el flanco de subida de la orden de control de escritura ( $\overline{W}$ ). Análogamente,  $\overline{EF}$  pasará a baja tras el flanco de bajada del control de lectura ( $\overline{R}$ ) durante el último ciclo de lectura que crea la condición de FIFO vacía.

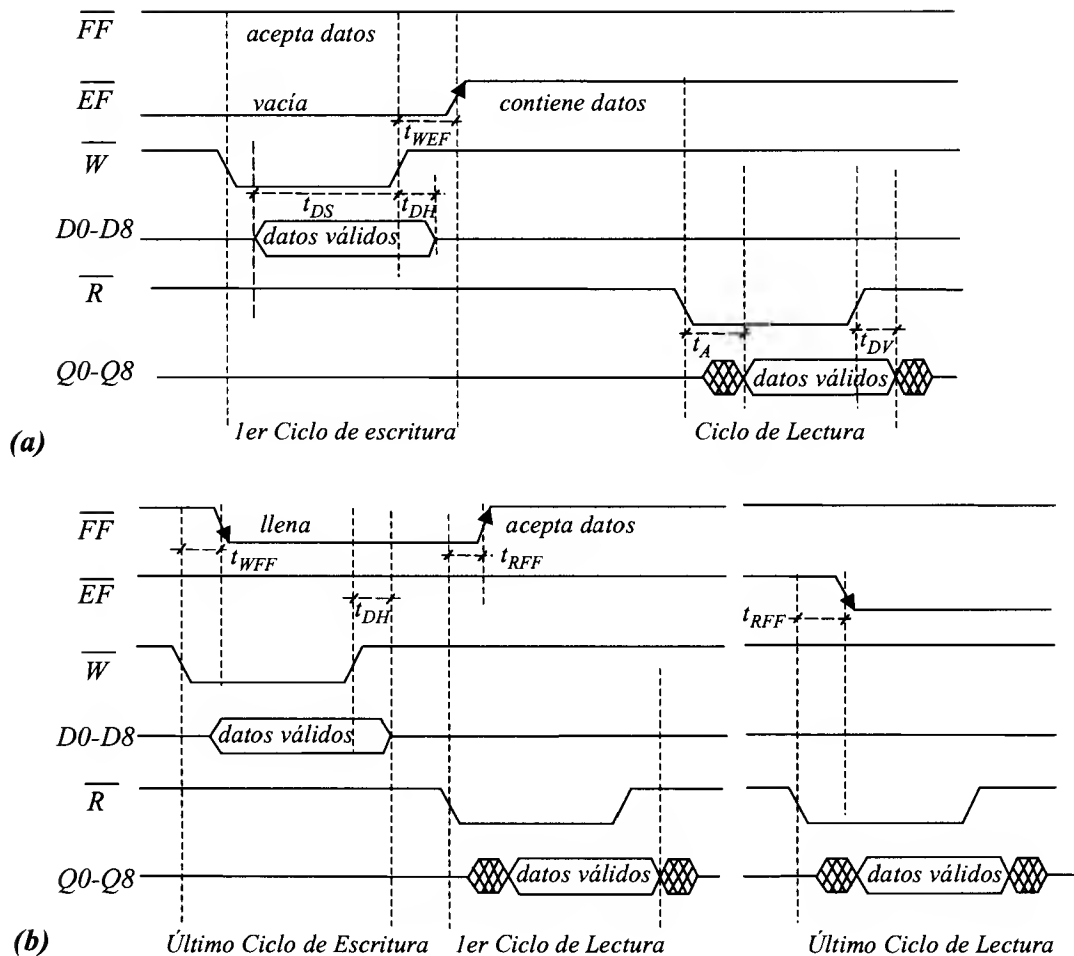
Para conseguir una mayor eficiencia en la comprensión del funcionamiento de este circuito es conveniente ver tanto el diagrama de bloques y la descripción funcional de sus terminales como el cronograma correspondiente a la evolución temporal de estas señales en los distintos modos de

comportamiento (ciclos de lectura y escritura) en distintas condiciones de los indicadores del estado de la FIFO. Así pues, conviene estudiar conjuntamente las *figuras 12.12 y 12.13*.



**Figura 12.12.** Diagrama de bloques de la FIFO Am7205A de AMD (8192x9).

El bit de condición  $\overline{FF}$  (*full-flag*) está también activo en baja. Mientras permanece en alta indica que la memoria no está llena del todo y por consiguiente pueden escribirse en ella nuevos datos. Cuando el puntero de la dirección de escritura está en una posición inferior que la del de lectura,  $\overline{FF}$  pasa a baja indicando que la memoria está llena e inhibiendo posteriores operaciones de escritura.



**Figura 12.13.** Cronograma de la FIFO. **(a)** Primer Ciclo de escritura seguido de un ciclo de Lectura. **(b)** Ultimo ciclo de escritura seguido del primer ciclo de lectura y finalmente del ultimo ciclo de lectura tras una serie de ciclos intermedios que han descargado la FIFO.

El terminal  $\overline{FL}/\overline{RT}$  (first-load/retransmit) se usa para pasar del modo de uso individual de la FIFO al modo de expansión en profundidad. El dispositivo que recibe primero los datos debe tener su  $\overline{FL}$  en baja y el resto lo tendrá en alta. Cuando sólo hay un dispositivo este terminal se usa para controlar la retransmisión del dato previamente escrito.

Los terminales  $\overline{R}$  (Read),  $\overline{W}$  (Write) y  $\overline{RS}$  (Reset) cumplen las funciones usuales ya

conocidas. Finalmente,  $\overline{XI}$  (*expansion in*) y  $\overline{XO}/\overline{HF}$  (*expansion-out/half-full*) controlan la posibilidad de conectar módulos iguales para expandir la FIFO en profundidad. Durante el funcionamiento como dispositivo único,  $\overline{XI}$  va a tierra y, cuando opera en modo extendido,  $\overline{XI}$  va al  $\overline{XO}$  del dispositivo previo. Análogamente, cuando  $\overline{XO}$  no se usa para conectar con el  $\overline{XI}$  del siguiente módulo, actúa como bit de condición para indicar que la FIFO está llena hasta la mitad.

La operación de la FIFO comienza con un ciclo de reset. Si  $\overline{FF}$  está en alta, la memoria puede recibir datos en un ciclo de escritura cuyo cronograma se muestra en la *figura 12.13.a*. La escritura comienza a la caída de ( $\overline{W}$ ) y los datos que aparecen en ( $D0...D8$ ) se graban de forma secuencial hasta que el bit de condición de memoria llena ( $\overline{FF}$ ), inhibe la escritura.

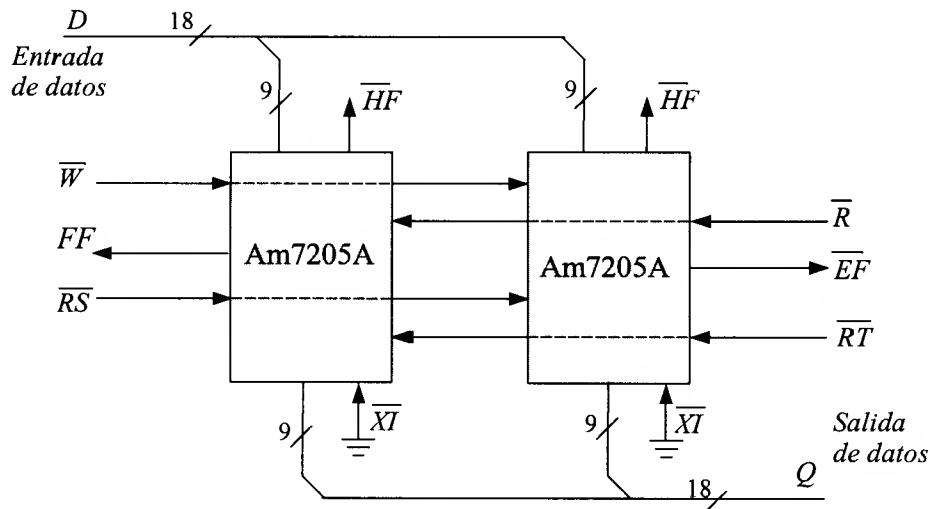
La operación de lectura se inicia con el bit de condición  $\overline{EF}$  en alta, indicando que la FIFO no está vacía y por consiguiente tiene sentido sacar datos de la misma. El ciclo de lectura se inicia con el flanco de bajada de la línea de control ( $\overline{R}$ ) (*Figura 12.13.b*). Los datos aparecen en ( $Q0...Q8$ ) y permanecen estables desde la caída a la nueva subida de ( $\overline{R}$ ). Entonces,  $Q0-Q8$  pasan al estado de alta impedancia, donde también permanecen cuando la FIFO está vacía o cuando opera en el modo de expansión pero no está activa.

Todos los circuitos de memoria (RAM, CAM, FIFO,...) son esencialmente elementos de diseño en sistemas digitales en general. Cuando el sistema digital del que estamos hablando es un sistema de cálculo estos circuitos cumplen las funciones que les pide la *Arquitectura de Ordenadores*. En el contexto de la *Electrónica Digital* nuestra función es describir el módulo y proponer procedimientos de análisis y síntesis. Cuando el circuito es de baja complejidad, podemos ser exhaustivos. Sin embargo, al aproximarnos al estado actual de la tecnología integrada el grado de complejidad de los circuitos que seleccionamos para desarrollar un tema concreto suele ser alto. Así nos pasó con las memorias CAM y así nos ocurre también con las FIFO tipo Am7205-A que estamos comentando.

En estas circunstancias, la descripción del significado funcional de los terminales, el cronograma de la evolución temporal de las señales de control ( $\overline{R}, \overline{W}, \overline{EF}, \overline{FF}$ ) en los modos de lectura y escritura y el tipo de conectividad necesaria para integrar a la FIFO en diseños más amplios, deben de seguir forzosamente las descripciones y especificaciones de las hojas de datos del fabricante correspondiente. Afortunadamente, existe un alto grado de coincidencia entre las estructuras internas,

el significado funcional de los terminales y los modos de operación de las distintas FIFO, pero esto no exime del uso directo de los datos reales correspondientes a cada solución concreta (por ejemplo, la solución TTL SN74ALS2239 ó el ejemplo de AMD que estamos usando aquí).

Volvamos ahora a nuestro ejemplo y veamos cómo se puede expandir en *anchura* y en *profundidad*. Recordemos que la expansión en anchura nos proporciona FIFOs con palabras de más bits y la expansión en profundidad nos proporciona más palabras del mismo número de bits. Las conexiones necesarias para estos dos tipos de expansiones se muestran en las *figuras 12.14* y *12.15*, adaptadas del catálogo de AMD ("MOS Data Book", 1991).



**Figura 12.14.** Composición de memorias FIFO para aumentar el número de bits. Paso de 9 a 18 bits usando dos circuitos Am7205A en paralelo. Los controles de lectura y escritura son comunes y el canal de datos se distribuye entre ambos circuitos.

En la expansión en anchura obtenemos una FIFO de  $8192 \cdot 9 \cdot N_1$  bits, haciendo comunes todas las líneas de control que entran a los  $N_1$  dispositivos. La conexión de la *figura 12.14* puede incluir ciertos elementos de azar al obtener los bits de estado ("status flags"). En las transiciones de alta a baja, se puede usar la función OR de los "flag" individuales porque no aparecen las desviaciones entre los comportamientos individuales. En cambio, cuando estos bits de condición realizan transiciones de baja a alta, pueden generarse valores falsos del bit compuesto ya que en estas circunstancias las desviaciones entre los comportamientos individuales si que son operativas. Si realizamos la



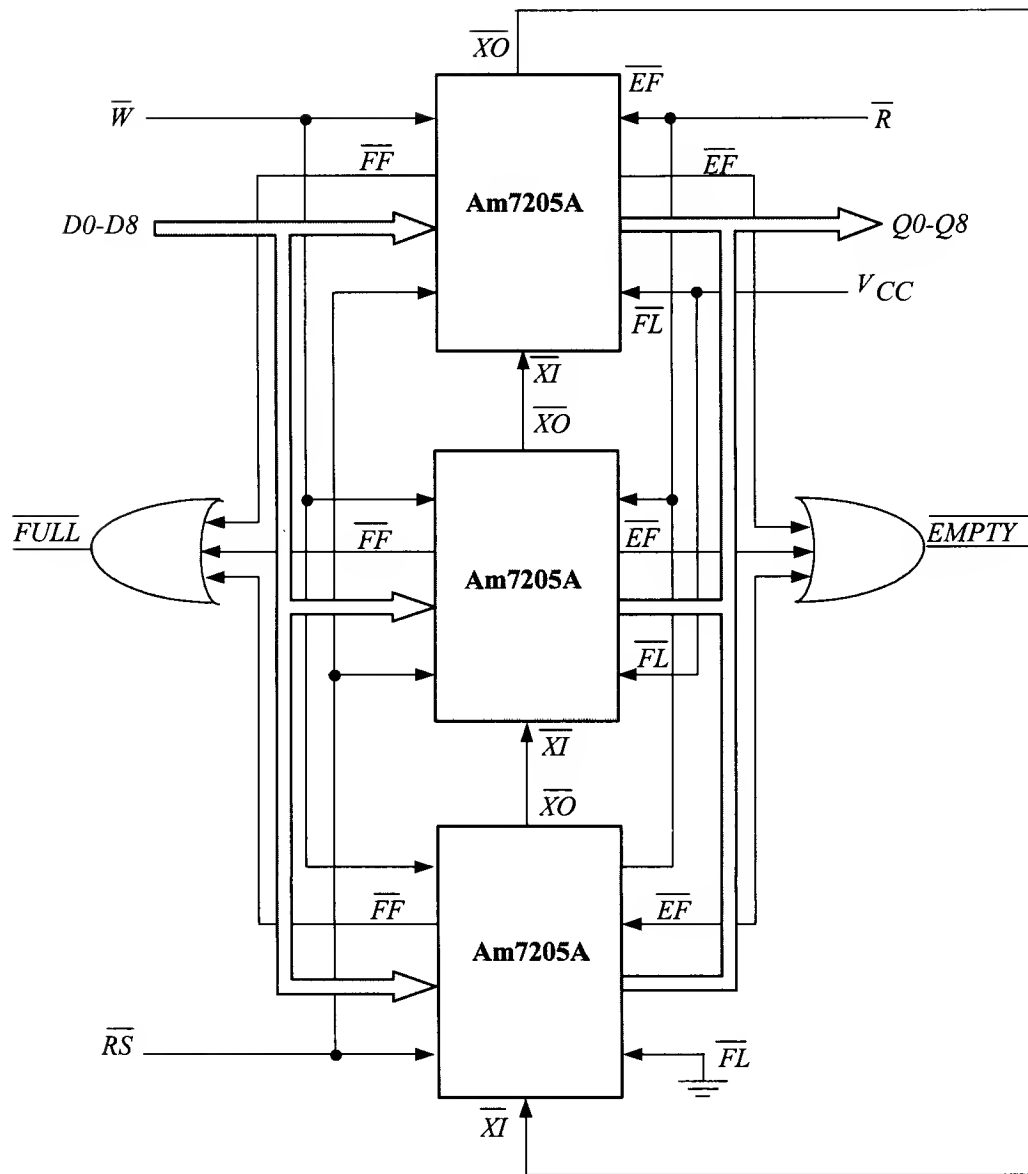
composición de los bits de condición mediante una puerta AND, tendríamos la situación inversa. Es decir, la transición de baja a alta no genera problemas, pero la transición de alta a baja produce "flags" erróneos. Para evitar ambos problemas se considera al "flag" de uno de los dispositivos como representante de la composición y los circuitos de control de lectura y escritura se diseñan para que operen en "el peor de los casos", cuando estamos seguros que se han superado los valores de los tiempos de asentamiento de ambos bits.

La *figura 12.15* muestra el circuito correspondiente a la expansión en profundidad. Es una conexión serie en la que los datos de entrada están accesibles (en paralelo) para todos los circuitos. El control se encarga de facilitar uno de ellos e inhibir todos los demás hasta que el circuito facilitado esté lleno. Entonces se inhibe este y se facilita el siguiente. Lo mismo ocurre en el proceso de lectura. En esta expansión obtenemos memorias de  $8192 \cdot 9 \cdot N_2$  bits que se configuran durante el ciclo de reset. La  $\overline{XO}$  de un dispositivo se conecta a la  $\overline{XI}$  del siguiente y la  $\overline{XO}$  del último cierra el lazo con la  $\overline{XI}$  del primero.

El dispositivo que va a recibir primero los datos tiene que tener su línea  $\overline{FL}$  (first load) en baja, mientras que los otros pueden tener esta línea en alta. Cuando se ha escrito en la última posición (8192) de un dispositivo activo, se produce una bajada en su pulso  $\overline{XO}$  y lo mismo ocurre cuando se lee en la última posición. La siguiente operación de  $\left[ \left( \overline{R} \right), \left( \overline{W} \right) \right]$  se realiza sobre el siguiente dispositivo que ahora estará activo, a la vez que inhiben todos los demás. En esta composición en profundidad necesitamos obtener la función OR de todos los bits  $\overline{FF}$  para saber cuando están llenos todos los dispositivos y análogamente con los bits de condición  $\overline{EF}$ , para saber cuando están vacíos.

Es posible también combinar ambas expansiones conectando en paralelo ristas de composiciones serie para obtener a la vez memorias FIFO de mayor longitud de palabra y de más capacidad.

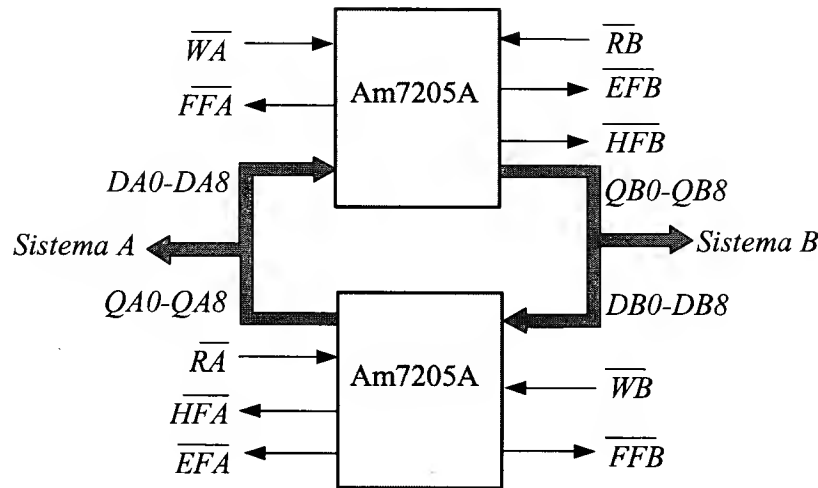
Existen muchas aplicaciones en las que nos interesa conectar dos sistemas (A y B) a través de sus FIFO mediante configuraciones bidireccionales y asíncronas. Existen soluciones integradas específicas para este tipo de problema como la Am4701 que es, además, programable. Sin embargo, siempre podemos conseguir una comunicación bidireccional usando circuitos unidireccionales como el que hemos usado en este apartado si los conectamos de la forma mostrada en la *figura 12.16*.



**Figura 12.16.** Expansión en profundidad de memorias FIFO. Se mantienen los 9 bits pero se aumenta el número de palabras.

Conectamos el sistema A con el B mediante una FIFO ( $A \rightarrow B$ ) en la que las entradas proceden de A ( $DA0...DA8$ ) y las salidas ( $QB0...QB8$ ) ya están en B. Inversamente, las salidas de la

FIFO (A → B) son las entradas de la FIFO (B → A) y las salidas de (B → A) son las entradas de (A → B).



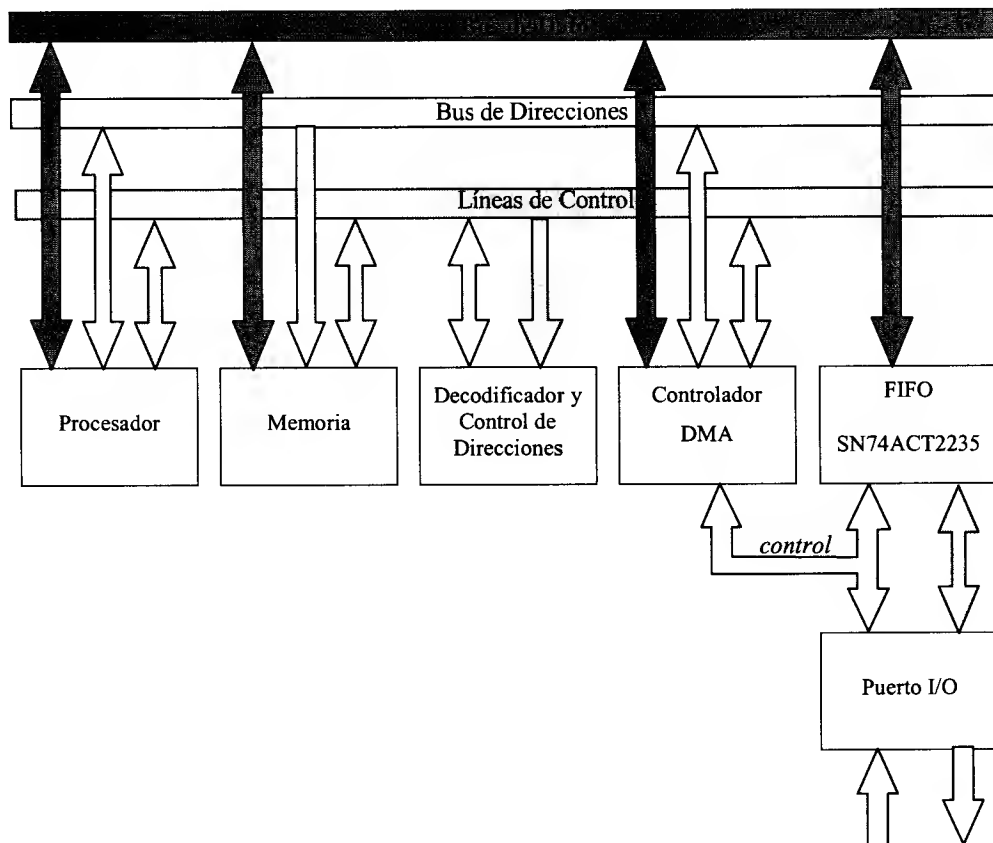
**Figura 12.16.** Configuración bidireccional.

Hay otras FIFO como la CY7C4215 de CYPRESS (522x18) cuya organización interna es análoga a la de AMD. Trabaja a 100MHz con tiempos de lectura/escritura de 10ns y posibilidad de lectura simultánea con la escritura. Su registro de estado posee cinco indicadores de nivel de carga (*vacía, llena, semi-llena, casi vacía y casi llena*). Los dos últimos son programables. La lectura y la escritura se controlan con relojes separados y los circuitos se pueden expandir usando los terminales para conexión en cascada de entradas y salidas y los correspondientes a la selección del dispositivo que carga primero. Tal como comentamos en la descripción general de las FIFO RAM de AMD, el bit indicador de memoria vacía sólo se activa bajo el control del reloj de lectura. Análogamente, el bit que describe el estado de memoria llena sólo se actualiza con el reloj de escritura. Esta arquitectura de indicadores sincronizados con relojes independiente nos garantiza que su descripción del estado permanecerá válida entre un ciclo de reloj y el siguiente.

Texas Instruments ofrece también memorias FIFO síncronas (SN74ALVC7813, 64x18), y bidireccionales (SN54ABT7820, 512x18x2) formadas a partir de dos FIFOs separadas y en tecnología avanzada BiCMOS.

## 12.6. APLICACIONES DE LAS FIFO

Ya hemos comentado que las aplicaciones de las FIFO están asociadas, en general, a su carácter de memoria “intermediaria” entre sistemas digitales de distinta velocidad o de modos diferentes de procesar los datos. Situaciones de este tipo se presentan, por ejemplo, en la conexión de periféricos a procesadores, en la conversión analógico-digital de señales de video, en la transferencia de datos por bloques o en la recolección de datos previos o posteriores a la ocurrencia de un determinado proceso. La *figura 12.17* muestra la conexión con el procesador de un periférico bidireccional usando una FIFO tipo SN74ACT 2235 de TI.



**Figura 12.17.** Conexión de un periférico bidireccional usando una FIFO tipo SN74ACT2235. La parte matizada corresponde al caso de periféricos con controlador de acceso directo a memoria. En este caso los datos son primero recolectados por la FIFO y después transferidos en bloque a la RAM.

Este esquema es representativo, a nivel cualitativo de un conjunto amplio de aplicaciones de la FIFO. Como los procesadores son más rápidos que sus periféricos, la función de la FIFO es acoplar esas velocidades diferentes. Por otro lado, y dado que una gran parte de los periféricos son bidireccionales, como un puerto paralelo o serie o un controlador de disco duro, es conveniente usar FIFOs bidireccionales que incluyen en el mismo circuito dos FIFOs unidireccionales independientes.

La FIFO puede actuar como buffer de entrada o de salida y en situaciones en las que se dispone, o no, de un controlador de acceso directo a memoria (DMA). Cuando la FIFO opera como un buffer de entrada y el procesador puede leer esos datos se genera una señal de interrupción condicionada por los indicadores del estado de carga de la memoria (*HALF FULL*, *ALMOST FULL* ó  *$\overline{FULL}$*  ).

El procesador opera de forma análoga en el proceso de escritura de datos en un periférico a través de la FIFO. Además, si combinamos la FIFO con un circuito controlador del acceso directo a memoria, podemos conseguir la transferencia de datos en bloques. Primero los acumula y cuando alcanza el nivel de alguno de sus indicadores de estado, *ALMOST FULL* ó  *$\overline{FULL}$* , los transfiere a la RAM del controlador del acceso directo a memoria, por ejemplo. Para ello se requiere que el control del bus durante la transferencia esté gestionado desde la FIFO. Cuando el paquete de datos ha llegado a la memoria RAM, se devuelve el control al procesador. Es decir, cuando se recibe una petición de transferencia de datos, el controlador del acceso directo a memoria (DMA) interrumpe al procesador y toma el control del bus, transfiere el paquete almacenado transitoriamente en la FIFO y devuelve el control al procesador. Si no usáramos esta FIFO, el procesador debería esperar hasta el final de cada transferencia individual.

Otro grupo de aplicaciones de las FIFO tiene que ver con la *adquisición de datos* para su posterior tratamiento digital o transmisión. Por ejemplo en tratamiento digital de imágenes o en videoconferencia donde una imagen de video de 1024x1024 pixels, por ejemplo genera 1Mbyte de datos por cada imagen. Tras una compresión de 32 a 1 todavía tenemos 32Kbytes de datos comprimidos que hay que procesar y/o transmitir. Para hacer más eficiente estos procesos, los datos se almacenan transitoriamente en una FIFO para esperar el momento de la transmisión o proceso. El resto del circuito digital puede entonces dedicarse a otra tarea, por ejemplo a adquirir y comprimir la siguiente imagen. La *figura 12.18* muestra de forma cualitativa este tipo de aplicaciones.

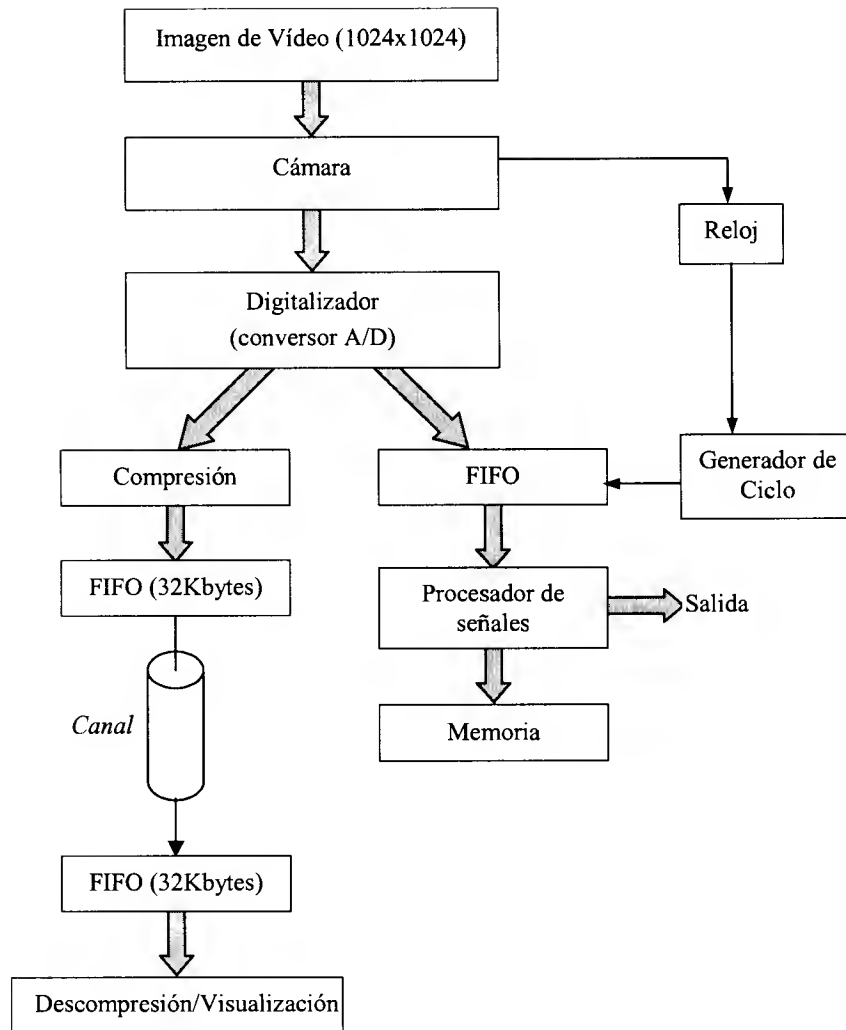


Figura 12.19. Esquema cualitativo de las aplicaciones de las FIFO en los procesos de adquisición de datos para su posterior tratamiento digital o transmisión.

## 12.7. PROBLEMAS

Hemos visto en este capítulo las memorias de acceso secuencial, los tipos de organización más usuales y la síntesis correspondiente mediante celdas RAM y estructuras CCD.

Para explicar el funcionamiento de estas memorias nos hemos apoyado en circuitos reales representativos del estado actual de la tecnología integrada. En estos ejercicios vamos a proponer la síntesis de estructuras mínimas que permitan ilustrar el funcionamiento de las memorias de acceso secuencial sin la necesidad de abordar la complejidad de los circuitos reales. Incluiremos también algunas cuestiones que ayuden a reflexionar sobre el funcionamiento de las estructuras CCD.

**E.12.1.** *¿Por qué decimos que las estructuras de los registros de desplazamiento en CCD pueden considerarse como transistores MOS multipuerta?. Proponer un circuito con transistores MOS en tecnología discreta que realice la misma función secuencial que el circuito mostrado en la figura 12.6.*

**E.12.2.** *Dibujar la evolución temporal de la tensión en el punto  $P_3$  (drenador del transistor  $Q_4$ ) en el circuito de la figura 12.4 cuando en la entrada,  $v_i$ , aparece la secuencia 110101. Asociar el valor de esta tensión con el correspondiente a los pulsos  $\Phi_1$  y  $\Phi_2$  del reloj bifásico que controla el desplazamiento.*

**E.12.3.** *En la figura 12.5 se ha propuesto un esquema de una etapa de registro en CMOS sin desarrollar mucho la explicación de su funcionamiento, basado en el funcionamiento de las puertas de transmisión mandadas por un reloj monofásico ( $\Phi$ ) y su señal complementaria ( $\overline{\Phi}$ ) y en los inversores CMOS que estudiamos en el tema 4.*

*Apoyándonos en las figuras 4.12 y 4.16 del tema 4, explicar el funcionamiento de la figura 12.5. Dibujar el cronograma de la evolución de las señales en  $I_1$ ,  $I_2$  y  $v_o$ , cuando en la entrada de la etapa,  $v_i$ , aparece la secuencia 010010. Usar como referencia las señales de reloj  $\Phi$  y  $\overline{\Phi}$ .*

**E.12.4.** *En la figura 12.3 se muestran tres organizaciones típicas de memorias secuenciales. a) Organización serie. b) Organización por lazos con direccionamiento directo para cada lazo y secuencial dentro del lazo y c) organización serie-paralelo-serie.*

*Diseñar los circuitos de control correspondientes para una versión mínima de cada una de estas organizaciones, sin considerar los mecanismos de refresco.*

**a)** *Se dispone de cuatro biestables D, puertas lógicas y circuitos de tiempo tipo 555. Proponer las conexiones adecuadas para entrada y salida de datos y control del desplazamiento en una memoria secuencial de 4 bits con organización serie.*

- b) *Partiendo de los mismos circuitos del apartado anterior diseñar una organización con dos lazos de 2 bits cada uno, incluyendo junto a los mecanismos de entrada, salida y desplazamiento, los correspondientes al decodificador de las direcciones de los dos lazos.*
- c) *Usando de nuevo los circuitos diseñados en los apartados anteriores y 4 biestables D adicionales, diseñar la versión mínima de la estructura serie-paralelo-serie. Es decir, un registro de 2 bits, dos vías paralelas de dos bits cada una y otro registro serie de salida, también de 2 bits.*
- E.12.5.** *En la figura 12.12 se muestra el diagrama de bloques de una memoria FIFO integrada para 8192 palabras de 9 bits sobre una estructura RAM con posibilidad de lectura y escritura síncrona.*
- Manteniendo todos los bloques funcionales (control de lectura, control de escritura, memoria RAM, punteros, bit de condición, lógica de reset y lógica de expansión), diseñar una versión mínima para cuatro palabras de dos bits. ¿Cómo podríamos conectar en serie y en paralelo dos FIFOs mínimas de este tipo?. Considerar el procedimiento propuesto en las figuras 12.14 y 12.15 para circuitos reales más complejos.*
- E.12.6.** *En la figura 12.2 aparece la estructura general de las memorias LIFO. Diseñar una versión mínima para dos registros de cuatro bits cada uno.*





---

# ***Preparación de la Evaluación***

---

## **Objetivos**

- 1: Organización de acceso secuencial*
- 2: Etapas dinámicas en MOS, CMOS y CCD*
- 3: Funcionalidad y arquitecturas de las FIFO*
- 4. Ejemplos y aplicaciones*

## **1. Organización de Acceso Secuencial**

- 1.1. El acceso secuencial completa los distintos modos de acceso a direcciones de memoria vistos en este texto. ¿Podría construir un cuadro resumen comentando las características distintivas, las ventajas y los inconvenientes de cada uno de los modos de acceso que conoce?.*

*El propósito de esta cuestión de evaluación es darle la oportunidad de que reflexione sobre su conocimiento global sobre el tema del direccionamiento de memorias.*

- 1.2. Las figuras 12.1 y 12.2 de este tema muestran las organizaciones FIFO y LIFO. Intente ilustrarlas para el caso más simple posible: Dos registros de desplazamiento de dos bits cada uno. ¿Cómo se controlan los procesos de lectura y escritura?. Use los ejercicios resueltos del tema 9 para ayudarse.*

## **2. Etapas Dinámicas en MOS, CMOS y CCD**

- 2.1. Explique cualitativamente el funcionamiento de las estructuras CCD. Use la analogía con transistores discretos propuesta en el ejercicio E.12.1. ¿Entiende el proceso de transporte de carga controlado por barreras de potencial?. ¿Ve la necesidad de usar relojes polifásicos?. Revise la figura 12.7 para ayudarse de un gráfico en la contestación a estas cuestiones.*

## **3. Funcionalidad y Arquitectura de las FIFO**

- 3.1. ¿Cuáles son las ventajas e inconvenientes asociadas al uso de celdas SRAM para las memorias FIFO?.*

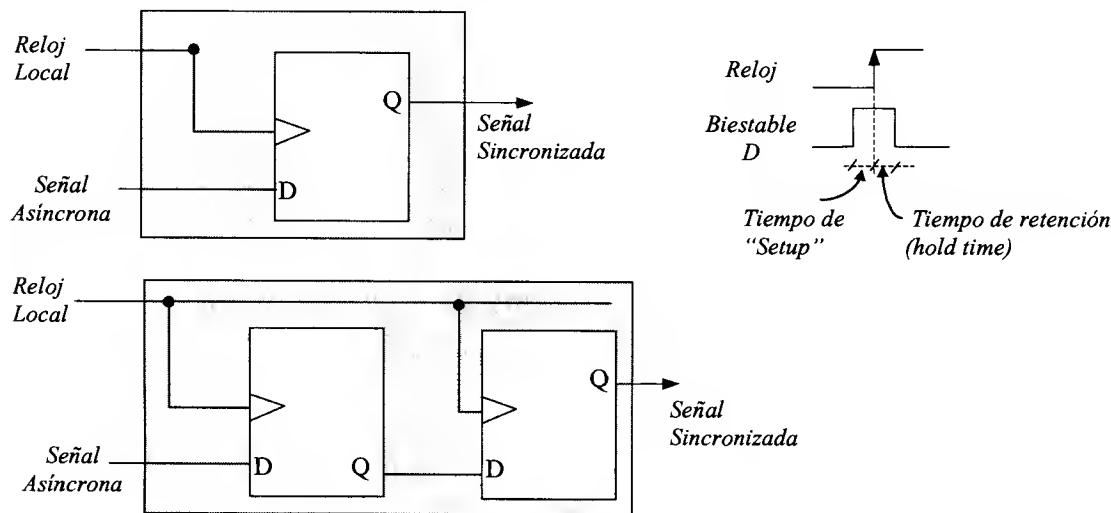
3.2. Describa las características funcionales de los terminales necesarios para la gestión de una FIFO asíncrona.

3.3. Dibuje el cronograma correspondiente a los ciclos de lectura y escritura de una FIFO asíncrona con  $\overline{CLEAR}$ ,  $\overline{FULL}$  y  $\overline{EMPTY}$  a la que sólo le caben tres palabras.

Suponga que primero, tras inicializar la FIFO con  $\overline{CLEAR}$ , tiene que escribir las palabras D1, D2 y D3. Después, el cronograma debe representar la lectura de D1 y D2 y, finalmente, el cronograma debe representar la escritura de D4 y D5. Comente el significado funcional de las transiciones de  $\overline{FULL}$ .

3.4. Describa las situaciones en los indicadores de estado que pueden dar lugar a problemas de inestabilidad en las FIFO asíncronas. Recuerde que los cambios de estado en  $\overline{FULL}$  y  $\overline{EMPTY}$  tienen que dar tiempo al asentamiento de las señales en torno a una transición del pulso de reloj.

3.5. ¿Cree que circuitos del tipo de los de la figura adjunta ayudan a resolver el problema de inestabilidad mencionado en la cuestión anterior?. ¿Por qué?.



3.6. ¿Podría describir los terminales adicionales necesarios para pasar de una FIFO asíncrona a una síncrona?.

- 3.7. *Describe el cronograma correspondiente a un proceso de escritura, seguido de otro de lectura y, finalmente, un tercero de escritura en una FIFO síncrona a la que sólo le quepan 4 palabras. Suponga, por ejemplo, que escribe cuatro, lee dos y vuelve a escribir una. Comente la evolución de los indicadores de estado ( $\overline{EMPTY}$  y  $\overline{FULL}$ ) a lo largo del proceso. recuerde que el cronograma debe empezar con la inicialización de la FIFO a través de  $\overline{CLEAR}$*
- 3.8. *¿Qué diferencias encuentra entre una FIFO organizada como registro de desplazamiento convencional y otra organizada circularmente con dos punteros?. ¿Hasta qué punto es adecuado llamar secuencial al direccionamiento de la segunda?. ¿Por qué?.*



Los ejemplos que hemos usado en el tema proceden de AMD, CYPRESS y TEXAS INSTRUMENTS.

## REFERENCIAS BIBLIOGRÁFICAS

1. Advanced Micro Devices: “*CMOS Memory Product*”. 1991.
2. Cypress Semiconductor: “*High Performance Data Book*”. 1995..
3. Texas Instruments: “*FIFO Architecture, Functions and Applications*”. Nota de Aplicación SCAA042A, 1999.
4. [www.music-ic.com](http://www.music-ic.com)
5. Rabaey, J.M: “*Digital Integrated Circuits. A Design Perspective*”. Prentice Hall Inc. 1996,
6. Weste, N.H.E., Eshraghian, K.: “*Principles of CMOS VLSI Design*”. Addison-Wesley, 1994.



---

## ***Lógica Secuencial Programable: CPLDs y FPGAs***

---

### **+ CONTEXTO**

*En el tema 7 hemos visto, esencialmente, la parte combinacional de la lógica programable, las PROMs, PALs y PLAs, lo que usualmente se conoce como SPLDs. Es decir, circuitos sencillos de lógica programable en los que la densidad de integración es baja, entre 100 y 500 puertas, y empaquetadas con un número de terminales entre 20 y 44. Los PAL 22V10 son representativos del estado actual de los SPLDs y, de hecho, se han convertido en estándares industriales, fáciles de diseñar, programar y usar.*

*Los calificativos de “sencillos” y “combinacionales” no son ciertos del todo porque ya incluyen macroceldas de salida con circuitos biestables que los hacen adecuados para aplicaciones en control lógico de bajo voltaje (3,3V) y bajo consumo. En realidad, estos SPLDs son idóneos para*

*el diseño de máquinas de estados finitos de hasta 2<sup>n</sup> estados, siendo n el número de macroceldas, y como tal se usan en telefonía celular y otros diseños portátiles. En este tema vamos a estudiar los CPLDs (Complex Programmable Logic Devices) y los FPGAs (Fields Programmble Gate Arrays) que corresponden esencialmente a circuitos lógicos programables de mucha mayor densidad de integración y con arquitecturas más flexibles, sin la limitación de los SPLDs en los que toda la lógica combinacional estaba limitada a las matrices AND y OR iniciales y toda la parte secuencial a los registros de las macroceldas finales.*

## **++ CONOCIMIENTO PREVIO NECESARIO**

*Todo el conocimientro necesario para la comprensión de este tema se ha visto en temas anteriores. En particular, las referencias al uso de fusibles o tecnología EEPROM para programar las CPLDs se ha visto en el tema 7. Las referencias al uso de celdas de memoria SRAM para reconfigurar la conectividad también han sido estudiadas en el tema 11 y lo mismo ocurre con las referencias a memorias FIFO o a máquinas de estados finitos.*

## **+++ OBJETIVOS DEL TEMA**

*El objetivo general de este tema es obtener una visión global del campo de la lógica secuencial programable, su frontera con los PLDs del tema 7 y algunas ideas sobre las tendencias actuales en este campo. De forma más concreta pretendemos:*

**Objetivo 1:** *Conocer la estructura interna, los módulos básicos, los procedimientos de interconexión y programación de los circuitos CPLDs.*

**Objetivo 2:** *Lo mismo para FPGAs con un intento de establecer sus fronteras con los CPLDs, que cada vez son más difusas.*

## **++++ GUÍA DE ESTUDIO**

*La extensión, complejidad y dinamismo del conocimiento y la tecnología electrónica asociados a la lógica programable hace que el carácter de este tema sea esencialmente informativo, con referencias a otros textos o direcciones de la red donde el lector interesado podrá encontrar mucha*

*más información. No obstante, hemos procurado sintetizar los aspectos de los distintos bloques funcionales y de la estructura de interconexiones en los dos grupos dominantes en el campo de la lógica programable de alta densidad, haciendo referencia a circuitos concretos de distintas casas comerciales que son representativos del estado actual de la tecnología electrónica.*



---

# ***Contenido del Tema***

---

## **13.1. ASPECTOS GENERALES DE LOS PLDs DE ALTA DENSIDAD**

Un circuito lógico programable (PLD) es un tipo de circuito electrónico que contiene, integrados en distintas arquitecturas, un número relativamente elevado de módulos lógicos idénticos configurables de acuerdo con diferentes esquemas de conectividad de los que emergen funciones terminales diferentes. Así, disponemos de una amplia variedad de circuitos secuenciales usando para su síntesis el mismo tipo de circuito integrado.

El carácter programable hace referencia al hecho de que, tanto la configuración interna de cada uno de los módulos como la conectividad de estos módulos entre sí y con los terminales de entrada y salida, pueden configurarse en el entorno de cada aplicación, de acuerdo con las especificaciones funcionales de cada usuario final.

Algunas de las formas de programar las interconexiones las hemos visto ya al estudiar las PROM, EPROM y EEPROM. Otras, basadas en el contenido de celdas de memoria RAM estática, las veremos en este tema.

Bajo esta definición general de lógica programable se incluye una gran diversidad de circuitos que nosotros hemos incluido en dos grupos: PLDs simples, con un número de terminales entre 20 y 44 (ya estudiados en el tema 7) y PLDs de alta densidad que, a su vez, se clasifican en dos grandes grupos: *CPLDs* y *FPGAs*. Aunque sus fronteras son cada vez más difusas, es posible especificar los aspectos comunes a ambos grupos y algunas características diferenciadoras.

Es común en ambos casos el aumento importante en la *densidad* de integración. Es decir, en el número de puertas y macroceldas integradas y en el número de terminales del circuito. También es común la búsqueda de una mayor *flexibilidad* en los *módulos* y en las *interconexiones* para vencer así las limitaciones de las PLDs simples cuya arquitectura estaba limitada al agrupar toda la parte combinacional en las matrices AND y OR de entrada y dejar la programación para las macroceldas de salida. Ahora se amplía la capacidad de cálculo de las macroceldas y se las sitúa distribuidas por toda

el área del circuito. Además, se incrementan y se hacen también más flexibles los recursos de ruta programada de señales. Estos dos elementos básicos, módulos lógicos e interconexiones van a estar presentes en la evolución de toda la lógica programable y recibirán distintos nombres dependiendo de la serie y de la casa comercial específica. Por ejemplo, los módulos lógicos reciben nombres tales como “macroceldas”, “celdas lógicas”, “bloques lógicos programables”, etc.

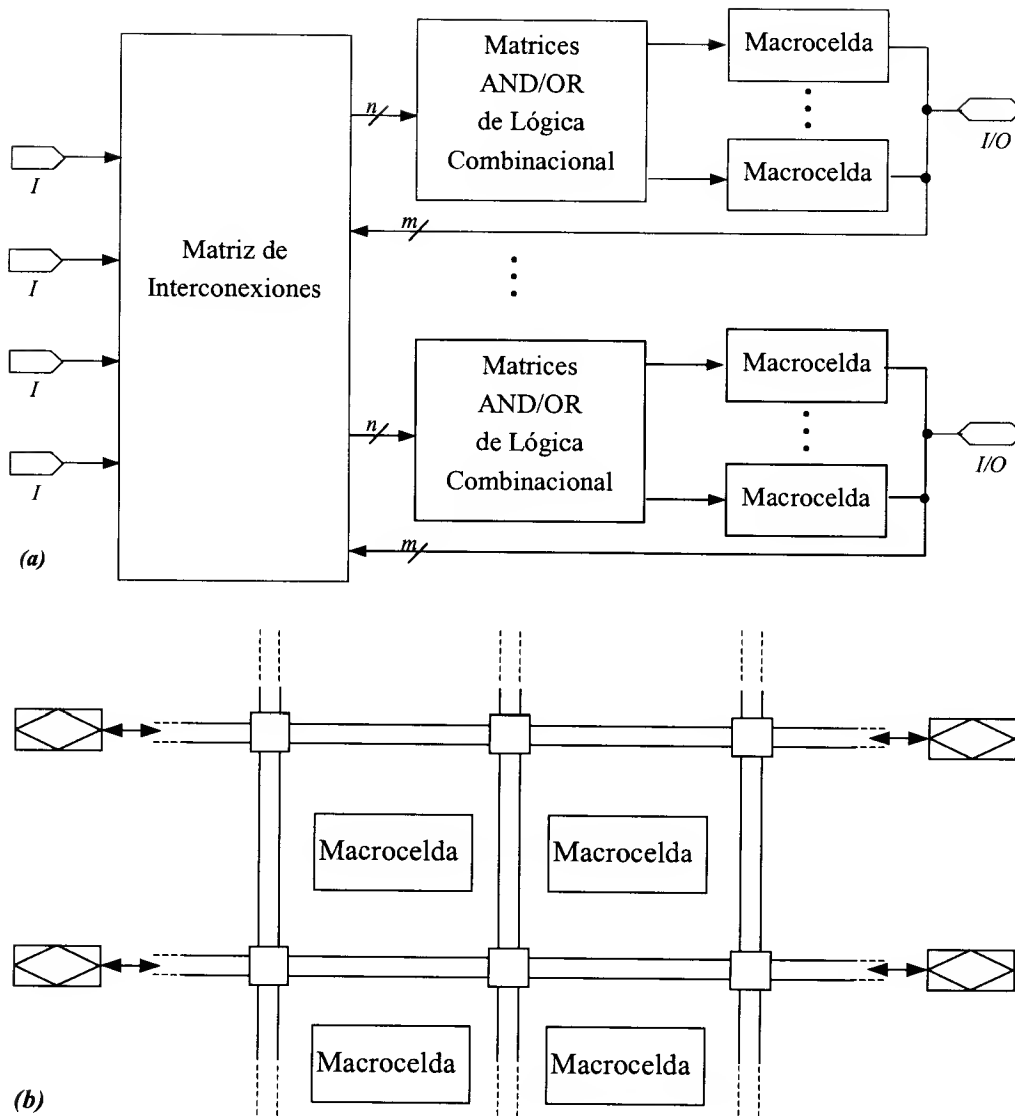
Veamos ahora algunas de las características que distinguen a los dos tipos de PLDs de alta densidad.

Los CPLDs tienen una *arquitectura segmentada*, como las de la *figura 13.1.a* formada básicamente por la yuxtaposición de un conjunto de PLDs simples, tipo PAL22V10, interconectadas a través de un esquema matricial de conexiones programables. Todas las entradas y todas las realimentaciones se conectan a los terminales producto (minterms) de todas las macroceldas. Además de los términos producto propios de cada macrocelda hay otros términos producto que son compartidos.

Estas arquitecturas CPLDs tienen tiempos de retardo predictibles y razonablemente constantes, independientemente de la función concreta para la que esté configurado el circuito. Esta configuración (programación) puede realizarse por métodos EEPROM o SRAM y cuando el circuito ya está integrado en el sistema final. Es decir, es no volátil y comienza a operar tan pronto como se conecta la fuente de alimentación.

En la parte *(b)* de la *figura 13.1* se muestra de forma esquemática la *arquitectura no segmentada*, característica de las FPGAs. Esta arquitectura no aprovecha el anterior nivel de integración (las PAL), sino que parte de una concepción modular de “grano” más pequeño. Ahora todo el cálculo (la parte combinacional y los biestables) se distribuye de forma homogénea sobre toda el área del semiconductor en forma de una matriz de “*bloques lógicos programables*” rodeada de bloques de entrada/salida. La interconexión ahora no se realiza en una matriz central sino que está distribuida sobre canales intercalados con los bloques , permitiendo así casi cualquier configuración de conectividad entre bloques. El ejemplo más representativo de la primera época de estas arquitecturas es el de XILINX, quien presentó en 1985 con el nombre de LCA (Logic Cell Array) su serie 4000. La serie XC-4000 consta de una matriz de CLBs (Configurable Logic Blocks), rodeada por un anillo periférico de módulos de entrada/salida, IOBs (Input-Output Blocks), e inmersa en una red de canales intermedios de conexiones programables.





**Figura 13.1.** Diagrama de bloques mostrando las diferencias en los esquemas de conectividad de los CPLDs. **(a)** Arquitecturas segmentadas, basadas en matrices tipo PAL. **(b)** Arquitecturas modulares, de grano pequeño y no segmentadas.

Hay varias características básicas en esta arquitectura:

- a) El incremento en la parte secuencial de los módulos lógicos que ahora están distribuidos de

forma homogénea. Cada CLB contiene cálculo combinacional y secuencial y puede configurarse para una amplia variedad de funciones.

- b) El total desacoplo entre los CLBs y los módulos de entrada y salida (IOBs) y el carácter programable de sus interconexiones.
- c) La naturaleza distribuida de las rutas que siguen las señales por canales intercalados entre las filas y columnas de CLBs y entre estos y los IOBs.
- d) La naturaleza de los elementos soporte de la programación. Antes teníamos fusibles y tecnología EEPROM y ahora añadimos SRAM, con un transistor de paso para definir en cada momento el estado funcional de cada punto de interconexión (PIP, Programable Interconnect Point). Así, los segmentos de conexiones internas están conectados o no, dependiendo del valor del bit almacenado en la celda RAM correspondiente al PIP de interconexión de esos segmentos.

Tal como comentamos previamente y como volveremos a comentar al final del tema, cada vez hay más elementos de hibridización de las arquitecturas y de los procedimientos de programación, buscando lo mejor de los fusibles, las EEPROM y las RAM. Por ejemplo, el problema de las SRAM es su volatilidad. Es decir, la pérdida de la configuración lógica al caer la fuente de alimentación, lo que obliga a cargar el programa de configuración desde una EPROM externa en cada proceso de inicialización. Esto aconseja combinar EEPROM con SRAM. También son difusas las fronteras de las FPGAs con los circuitos de aplicaciones específicas programables por máscara. La combinación de estas opciones da lugar a nuevos circuitos que acomodan funciones programables a nivel de sistemas, SPGAs (System Programmable Gate Array), donde se pueden configurar no sólo funciones combinacionales y secuenciales de carácter convencional como contadores, registros y controladores, sino que también pasan a ser programables funciones de memoria, rutas controladas de datos, procesadores aritméticos, etc...

En el resto de este tema vamos a presentar los aspectos básicos de la evolución de los CPLDs y los FPGAs. Un tratamiento en profundidad de la lógica programable, desde la tecnología electrónica hasta el desarrollo de aplicaciones, pasando por el uso de lenguaje de descripción del hardware tipo VHDL ó Verilog, queda fuera del alcance de este texto.

## 13.2. EVOLUCIÓN Y EJEMPLOS DE LAS ARQUITECTURAS CPLD

Ya hemos dicho que un CPLD consta esencialmente de unos cuantos bloques lógicos tipo PAL (matrices AND y OR y macroceldas) y un esquema de interconexión programable entre esos bloques y las celdas de entrada/salida. Tal como vimos en el tema 7, los bloques lógicos realizan sumas de productos y las macroceldas de salida almacenan en biestables, sacan al exterior y/o realimentan los resultados de estas funciones a través de los multiplexos. Así, pasar de PAL a CPLD es en realidad un problema de escala, de número de puertas y de número de terminales en el circuito. Hay tres números que definen los recursos de un CPLD y que, por consiguiente, determinan el carácter de simple o complejo de un PLD:

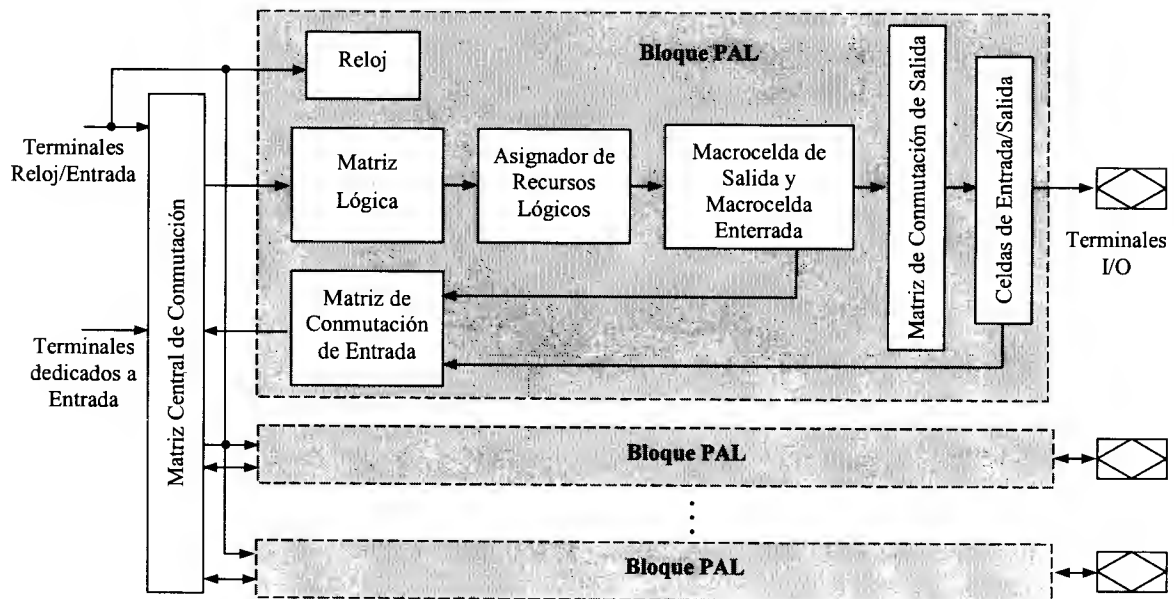
- 1) El *número de macroceldas* asociadas a cada bloque lógico. De él depende la capacidad de los registros que pueden configurarse para sintetizar contadores, por ejemplo. Este número es una cota superior del número de estados del autómata finito que puede sintetizarse con el CPLD. Evidentemente, para  $Q=2^n$  estados necesitamos  $n$  biestables  $D$ .
- 2) El *número de entradas* que van a cada bloque lógico desde la matriz de interconexiones.
- 3) El *número de términos producto* por bloque y la forma en la que puede realizarse el reparto. Por ejemplo, añadiendo a los minterms propios del bloque un conjunto de minterms compartidos. Al igual que con los biestables de las macroceldas, el número de minterms define también un límite superior a la complejidad de las funciones lógicas que puede acomodar el circuito. Veamos ahora algunos ejemplos.

### 13.2.1. Familias MACH de AMD/Vantis

Las familias MACH (Macro Array CMOS high) usan tecnología EEPROM para la programación de los conmutadores. Contienen en varias PAL interconectadas a través de una única matriz de conmutación tal como se muestra en la figura 13.2. Por ejemplo el circuito MACH231SP contiene 8 bloques PALCE 22v10. A esta matriz llegan señales de los terminales configurados específicamente como entradas de las celdas I/O de salida que pueden configurarse como entradas, como salidas o en estado de alta impedancia y de las realimentaciones procedentes de las macroceldas. Después, al programar la matriz de conmutación, estas señales son redireccionadas como entradas efectivas a las matrices AND/OR de las PAL para contribuir a la construcción de los terminales mínimos que participan en cada bloque PAL. En la *figura 13.2* hemos resumido todos los

elementos funcionales de cada bloque PAL en las siguientes componentes:

- 1) *Generador de términos mínimos*
- 2) “*Acomodador*” de esos términos mínimos, asociándolos en número variable a las distintas macroceldas.
- 3) *Macrocelas de salida.*
- 4) *Macrocelas enterradas.*



**Figura 13.2.** Versión simplificada de los componentes funcionales y de la estructura de cada uno de los bloques PAL que forman los CPLDs de la familia MATCH-4 de Vantis.

Como ya conocemos del estudio de las PAL, las macroceldas de salida pueden configurarse como combinacionales o secuenciales, a través de biestables D ó T. Las macroceldas enterradas que incluye la serie MACH2, permiten duplicar el número efectivo de macroceldas a efectos de cálculo interno y realimentación sin aumentar el número de terminales del circuito.

El circuito encargado de asignar los distintos términos mínimos los asocia, en grupos de cuatro, a las distintas macroceldas en función de las exigencias de cada diseño concreto. Así, por ejemplo, en el circuito MACH231SP cada bloque PAL dispone de 64 términos producto de 32 entradas externas y

el distribuidor se encarga de asignar esos 64 términos producto a las 16 macroceldas, de acuerdo con la configuración programada.

La familia MACH4 aumenta la densidad hasta 256 macroceldas y añade un reloj múltiple, lógica síncrona y asíncrona y opciones flexibles de set y reset. Cada bloque PAL tiene ahora su propio generador de reloj.

La familia MACH5 incorpora dos niveles de interconexión. Cada grupo de cuatro PAL pone sus propios recursos de interconexión. Después, estos grupos de cuatro PAL, llamados segmentos, entran en un segundo nivel de interconexión. Además, los circuitos pueden programarse y reprogramarse sin necesidad de extraerlos de la placa en la que se encuentran ensamblados con el resto de los circuitos de un sistema concreto (ISP, In System Programming). Así, el intervalo temporal entre la fase de diseño y la salida al mercado de una aplicación se reduce notablemente. Esta ventaja está asociada esencialmente al uso de tecnología EEPROM (no volátil) para el almacenamiento del programa de configuración.

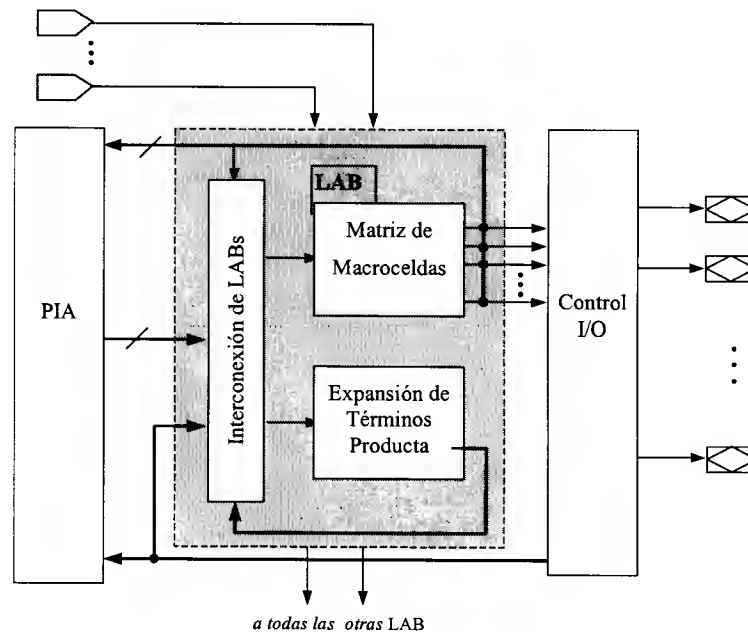
### **13.2.2. Familias MAX y FLEX de Altera**

La familia MAX (Multiple Array Matrix), con ligeras modificaciones, sigue la estructura segmentada vista anteriormente. También aquí las macroceldas se organizan en grupos llamados LAB (Logic Array Block) equivalentes a las PAL de antes y después esos LAB se interconectan usando PIAs (Programmable Interconnection Array) que son buses globales alimentados por todos los terminales I/O y por las macroceldas. También se incluyen terminales dedicados sólo a funciones de entrada. Lo distintivo de esta familia es:

1. La extensión de las macroceldas que permiten configurar sus biestables como D, T, R-S ó J-K con funciones de *preset* y *clear* e incluye un reloj programable de forma independiente.
2. La posibilidad de que las macroceldas compartan minterms, hasta un total de 32 por macrocelda. Para aumentar la flexibilidad se incluyen “expansores” que son grupos de minterms no asignados inicialmente a ninguna macrocelda.
3. Hay relojes globales y para los distintos grupos de macroceldas (LAB)

La *figura 13.3* muestra la arquitectura de la serie 5000. Esta serie llega hasta las 192 macroceldas por circuito. En la serie 7000, *Figura 13.4* la tecnología de programación pasa de EPROM a EEPROM y se aumenta la densidad hasta las 256 macroceldas permitiendo la

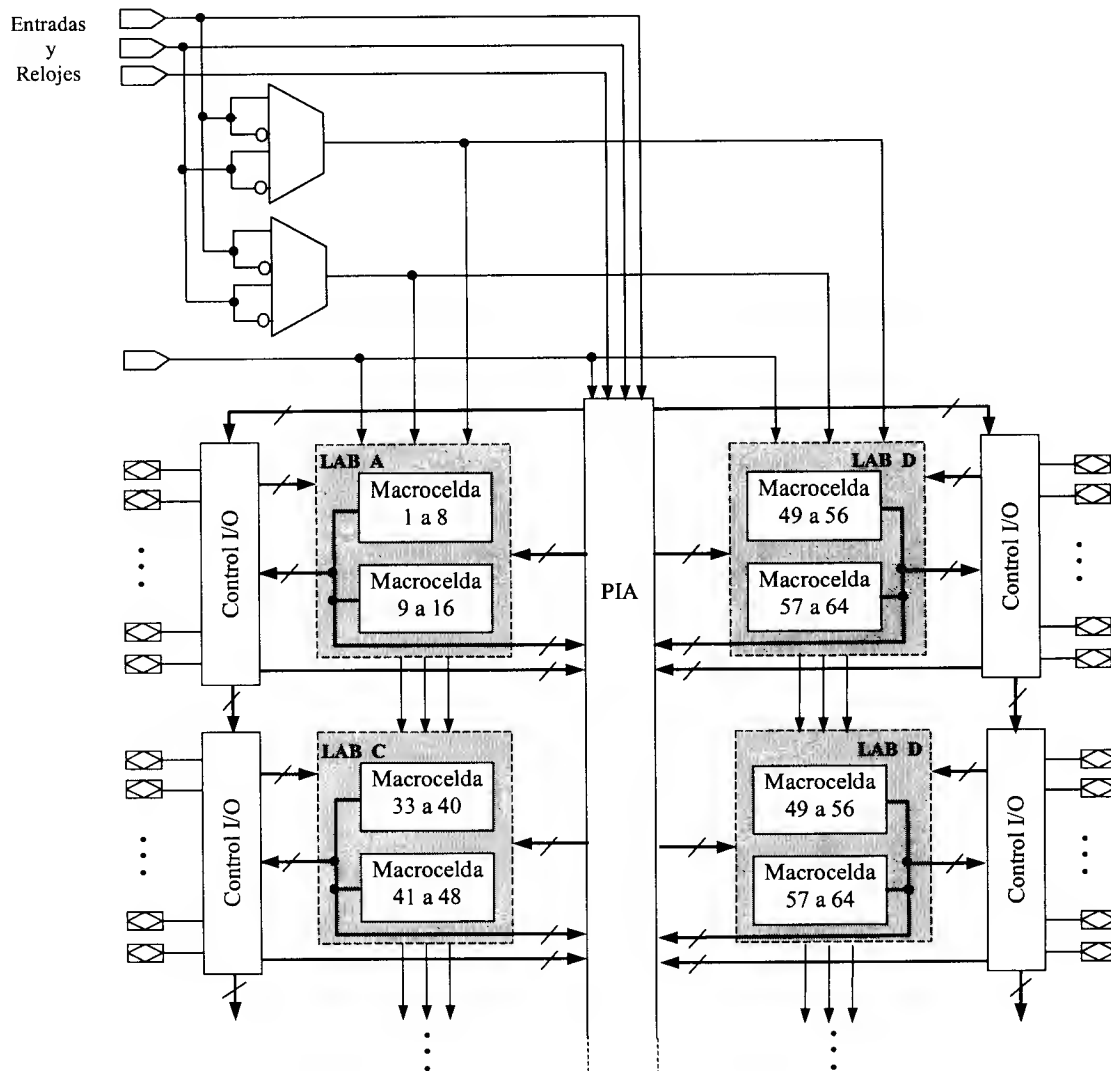
programación en la placa (ISP). Finalmente, la serie 9000 conserva las funcionalidades anteriores, aumenta el número de macroceldas hasta 560 y añade nuevas vías rápidas de interconexión programable (FastTrack Interconnect) para filas y columnas. Estas vías de conexión rápida están intercaladas entre los distintos LAB. A su vez, estos LAB mantienen su matriz local de interconexión entre las 16 macroceldas del grupo. Dentro del grupo existen también los mismos esquemas de realimentación desde los biestables de salida de cada macrocelda a la matriz de conectividad local, donde se combinan con las entradas externas, para generar los minterms de cada una de las macroceldas.



**Figura 13. 3.** Versión simplificada de la arquitectura de la serie MAX-4000 de Altera, donde focalizamos en uno de los LABs (grupos de macroceldas).

Si reflexionamos un poco sobre la evolución de los PLDs, vemos cómo se repite el proceso iterativo en el diseño y en la integración previsto por Hennie en 1961. Lo que buscamos esencialmente, el circuito secuencial universal y programable capaz de crecer de forma recursiva a través de redes de interconexión también universales y programables. Como la solución formal obvia que vimos en el tema 9 ( $N$  autómatas modulares universales con  $2^{N+M}$  entradas cada uno) no es tecnológicamente la más adecuada por la explosión combinatoria en el número de términos mínimos,

en el número de terminales necesarios y, sobre todo, en la dimensión de la matriz de interconexiones programables ( $2^{2^{N+M}} \times 2N$ ), se hacen necesarias una serie de estrategias para copar con esa dificultad.



**Figura 13. 4.** Visión más global, también simplificada de la arquitectura de la serie 7000.

La más general de estas estrategias ha sido la segmentación en las dos componentes estructurales básicas en todo CPLD:

- 1) *Los módulos lógicos.*
- 2) *La estructura de interconexiones.*

Así, primero se separa la parte combinacional de la secuencial en los módulos lógicos y después se combina en las macroceldas de salida. Más tarde se segmentan las arquitecturas agrupando un conjunto de macroceldas y generadores de términos mínimos para formar bloques tipo PAL o LAB. Finalmente se separan también las interconexiones programables distinguiendo la conectividad dentro de un grupo de las vías largas y rápidas de interconectividad entre grupos y de estos con los módulos de entrada y salida. Además, en todo este proceso evolutivo de los CPLDs se aumenta constantemente la densidad de integración. Es decir, el número de macroceldas por grupo (PAL ó LAB) y el número de grupos por circuito.

Otro paso en la evolución de las CPLDs es la arquitectura FLEX (Flexible Logic Element Matrix) de Altera donde de hecho se confunde la arquitectura CPLD con la propia de las FPGAs, porque combina elementos propios de una arquitectura segmentada (de grano grueso) con otros propios de una arquitectura homogénea (de grano pequeño), propia de las FPGAs donde ya no se distingue entre parte combinacional (la generación inicial de los minterms) y secuencial (las macroceldas) sino que se diseña en términos de un conjunto de elementos lógicos, LE (“Logical Elements”) que combinan parte combinacional y secuencial. Es decir, son una versión modificada de las macroceldas que ahora se distribuyen de forma homogénea sobre todo el circuito. Sin embargo, la arquitectura continua siendo segmentada porque estos nuevos elementos lógicos (LE) también están agrupados en LAB (Logic Array Blocks) para facilitar las interconexiones rápidas.

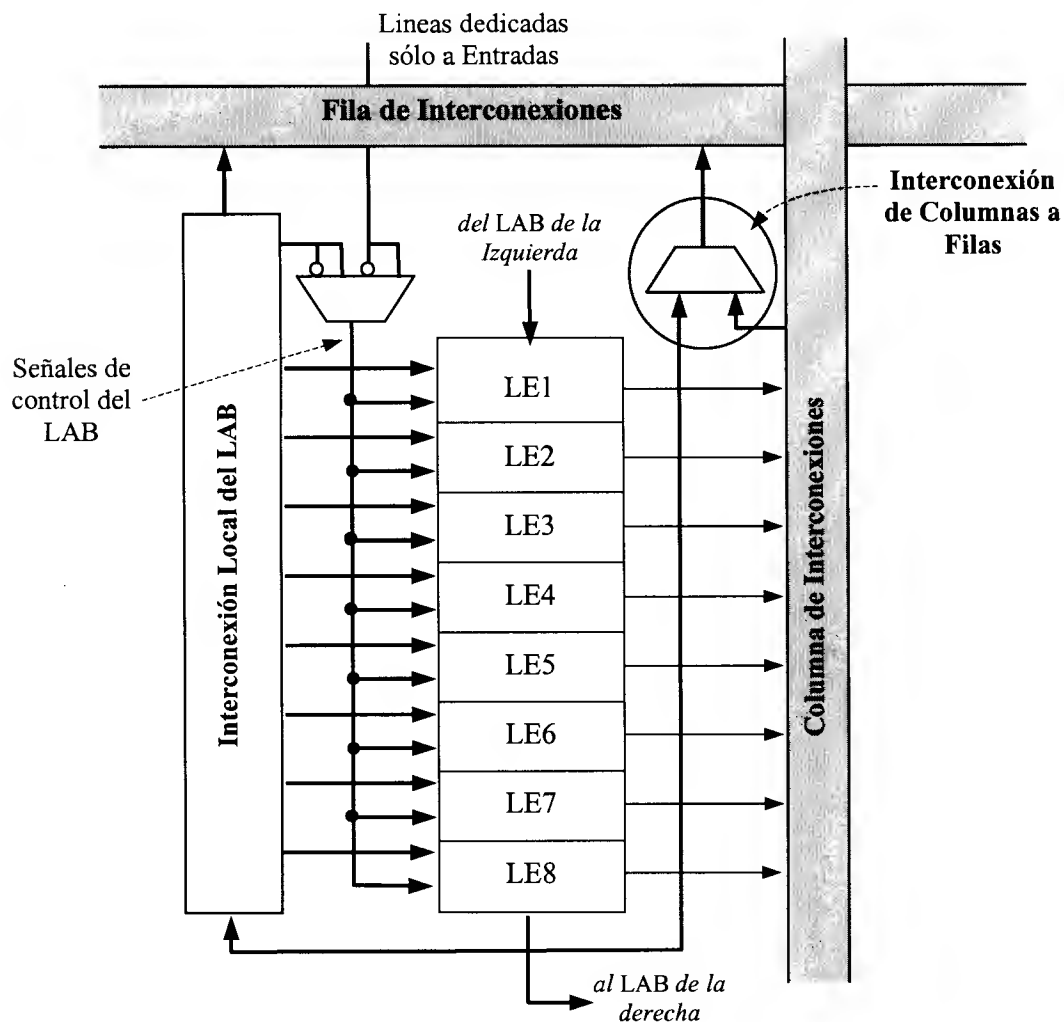
Dentro de las arquitecturas FLEX hablaremos de las series 8000 y 10K. La primera porque incorpora las características de series anteriores, como la 6000, y la segunda porque introduce elementos de novedad (bloques “empotrados”) que, de nuevo, la hacen diferente de las FPGAs.

Cada elemento lógico (LE) consta de una tabla de sólo lectura (LUT, “Look-Up Table”) de cuatro entradas, un biestable programable como D, T, J-K o R-S, con preset y clear, un reloj y entradas y salidas para facilitar el arrastre en la conexión en cascada de varios elementos. Estas tablas LUT son en realidad un procedimiento para obtener cualquier función lógica de cuatro variables por lectura del valor correspondiente a cada una de las 16 configuraciones mutuamente exclusivas para



cada una de las funciones.

La *figura 13.5* muestra el esquema de los LAB que constituyen el siguiente nivel de agrupamiento lógico que enlazan a 8 LE y se organizan en forma de matriz.



**Figura 13.5.** Diagrama de bloques simplificado de un LAB (Logic Array Block) de la serie FLEX 8000 de Altera. (las líneas de trazo grueso representan buses)

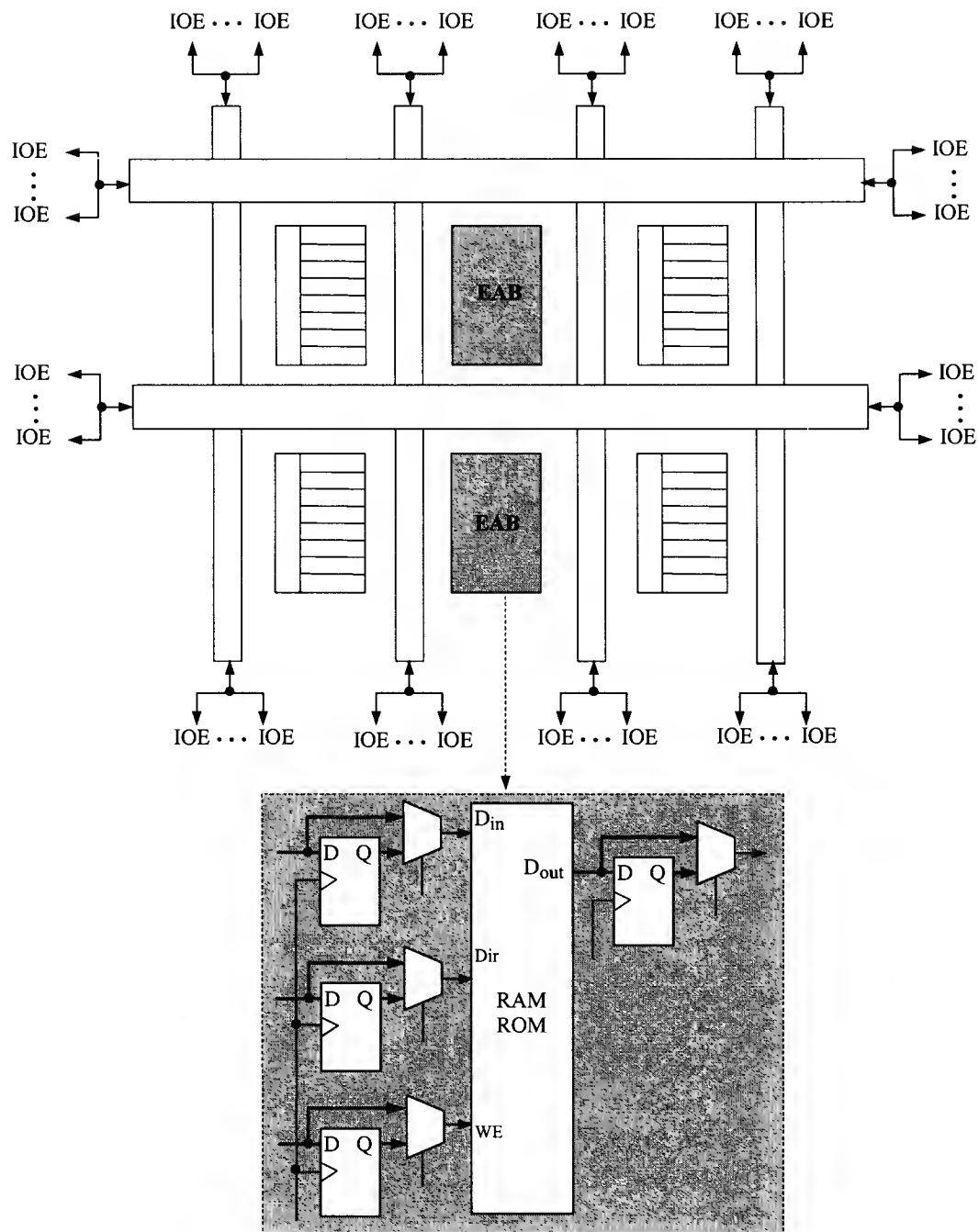
Dentro de cada LAB hay una red de interconexión local para sus 8 elementos lógicos. Además,

los distintos bloques LAB del CPLD se conectan a través de otras vías rápidas y largas que recorren las filas y columnas de la matriz. Cada una de estas filas y columnas facilita el intercambio de información de las LAB correspondientes con los elementos de entrada y salida (IOEs) que están al final de las filas y las columnas, en la periferia de la matriz. Estos IOE son buffers de tres estados.

Los elementos lógicos de la serie FLEX 8000 pueden operar en cuatro modos diferentes : *modo normal*, *modo aritmético*, *modo contador reversible* y *modo contador con clear*. En cada uno de estos modos se usan de forma diferente (orientada a un tipo de aplicación) los recursos combinacionales y secuenciales disponibles. El nombre de los modos es representativo de las aplicaciones a las que van dirigidos. Así, en el modo normal los LE se configuran para aplicaciones lógicas de propósito general. En el modo aritmético los LE se configuran en dos LUT de tres entradas que los hacen más adecuados para la síntesis de sumadores, acumuladores y comparadores. Una LUT genera funciones de tres variables y la otra se usa para generar el arrastre. En los otros dos modos los LE se configuran para facilitar la síntesis de contadores.

Veamos finalmente las características generales y la arquitectura de la familia FLEX 10K, basada en celdas SRAM en tecnología CMOS que son preconfigurables. Sus dos componentes principales son los LAB que ya conocemos y que se usan para implementar funciones combinacionales o secuenciales convencionales tales como sumadores o contadores o autómatas finitos y los nuevos bloques “empotrados” EAB (Embedded Array Block). Estos EABs se usan para implementar bloques de memoria, multiplicadores, microprocesadores o procesadores digitales de señales (DSP). Se pretende así que todo un sistema digital pueda configurarse en un CPLD y reconfigurarse sin desconectarse de la placa, cargando los nuevos datos de configuración. Como la reconfiguración requiere menos de 320ns, se pueden introducir cambios en tiempo real, durante el funcionamiento del circuito.

La *figura 13.6* muestra el esquema general de la familia FLEX 10K. En la parte superior se muestra el diagrama de bloques distinguiendo los bloques empotrados (EAB) de los otros elementos lógicos (EL) que ya conocemos. En la parte inferior de la misma figura se ilustra el esquema de un EAB que consiste, esencialmente, en un bloque de memoria configurable como RAM o ROM y como 256 palabras de 8 bit, 512 palabras de 4 bits, 1024 de 2 bits ó 2048 de 1 bit. La arquitectura es análoga a la de la serie 8000 con los dos tipos de vías de interconexión, las LUT de cuatro variables y el biestable programable en cada uno de los LE de cada LAB y las conexiones con los elementos de entrada y salida al final de cada canal rápido de interconexión entre LABs.



**Figura 13.6.** Esquema general de los circuitos FLEX 10K con bloques empotrados de memoria RAM/ROM

Lo realmente distintivo está en esos bloques adicionales, EABs, que incrementan notablemente la capacidad computacional del circuito. Veremos después que las FPGAs también incluyen pequeñas memorias RAM distribuidas pero estas EABs tienen ventajas frente a la RAM de los FPGAs porque aquí los bloques RAM son mayores con lo que disminuyen los retardos y, además, su valor es más predecible.

Los EABs incluyen biestables delante y detrás del núcleo RAM y pueden usarse también para generar funciones lógicas complejas. Para ello basta con programarlos durante la configuración del circuito como memorias de sólo lectura (ROM), dando lugar así a una gran LUT en la que podemos ir a leer el valor de las funciones de forma directa, en vez de configurar un circuito que las calcule. Esta implementación de funciones mediante tablas es más rápida porque minimiza el tiempo de acceso.

### **13.3. EVOLUCIÓN Y EJEMPLOS DE LAS ARQUITECTURAS FPGA**

Ya hemos comentado que la evolución de las arquitecturas CPLD y FPGA, hace que a ciertos niveles de complejidad tengan elementos comunes. Sin embargo, todavía hay dos factores que permiten la distinción:

1. Las FPGAs son arquitecturas flexibles y de “grano pequeño” que pueden contener más de 10.000 celdas lógicas idénticas, cada una de las cuales puede configurarse independientemente.
2. Las interconexiones entre los elementos se realizan a través de una matriz de conductores y conmutadores programables, en general a través de celdas de memoria RAM ó EEPROM.

La tecnología actual produce FPGAs en las que se llama “grano pequeño” a un par de transistores, algunas puertas, multiplexos o memorias de sólo lectura (LUTs) con biestables incluidos con lo que el “grano” ya no es tan pequeño, sino que se aproxima a la complejidad de las macroceldas vistas anteriormente.

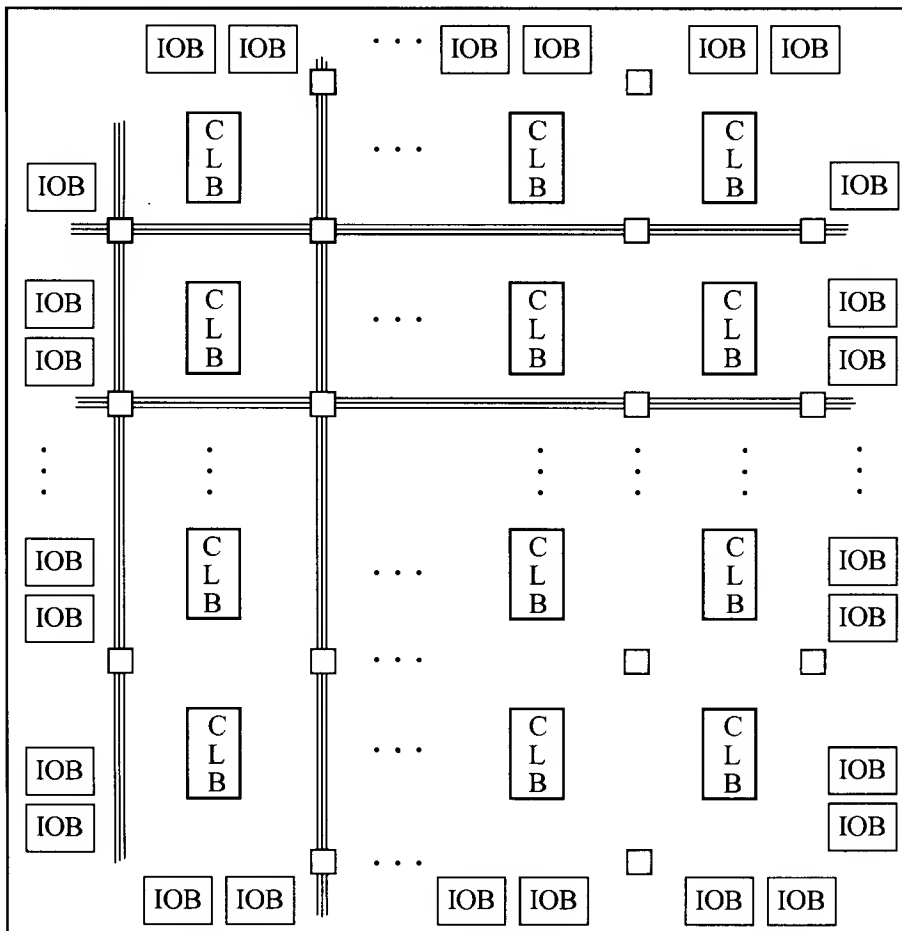
En cuanto a las interconexiones, dependiendo del número de segmentos conductores incorporados, también hay distintas arquitecturas. En general se busca una solución de compromiso porque, si hay pocos segmentos, no podremos utilizar todos los bloques lógicos y si hay muchos segmentos aumenta el tamaño del circuito y disminuye la eficiencia del mismo.

De los tres grupos en los que se clasifican usualmente las arquitecturas FPGA (organización por filas, organización simétrica y arquitecturas de grano pequeño), vamos a comentar sólo algunos

ejemplos de arquitecturas simétricas de grano medio, similar a las macroceldas, y con una jerarquía de redes de interconexión por filas y columnas con configuración controlada por memoria SRAM.

### 13.3.1. Familia XC4000 de XILINX

La familia XC4000 tiene una arquitectura regular y flexible basada en tres elementos: (1) una matriz de bloques lógicos configurables, CLB (Configurable Logic Blocks), (2) una red jerárquica programable de recursos de interconexión y (3) una periferia de bloques de entrada y salida, IOB, (Input/Output Blocks), tal como se muestra de forma esquemática en la *figura 13.7*.



**Figura 13.7.** Aspecto general de la arquitectura regular y programable de la familia XC-4000 de XILINX.

Los tres tipos de elementos (CLG, IOB e interconexiones) son configurables cargando los datos de configuración en la memoria RAM interna desde una EPROM externa al conectar la alimentación general del circuito, a través de un comando, o bien desde un sistema de cálculo (PC o estación de trabajo). Existen entornos de desarrollo y software de simulación para la captura de esquemas y la realización del diseño que termina configurando la función local de cada uno de los CLB's y IOB's, junto con sus conexiones.

En la primera generación de LCA's, introducida en 1985, los bloques lógicos incluían funciones combinacionales de cuatro variables y un biestable. En la segunda generación (1987), los CLB's se extienden para incorporar funciones lógicas de más variables y un segundo biestable. La tercera generación expande esta arquitectura incluyendo más biestables, haciendo más flexibles los generadores de funciones lógicas, aumentando la memoria RAM incluida en el circuito y haciendo más potentes las interconexiones permitiendo seis modos de programación. La serie XC4000 es representativa de esta generación y ofrece hasta 900 CLB's (30·30), 28K de RAM, 240 bloques I/O y 20.000 puertas. Las nuevas series XC-4000E y XC-4000x extienden las funcionalidades de la familia XC-4000 aumentando la capacidad y la velocidad de los circuitos. Incluyen también elementos nuevos en la arquitectura tales como nuevas facilidades para ruta de datos, buffers de reloj de alta velocidad, etc...

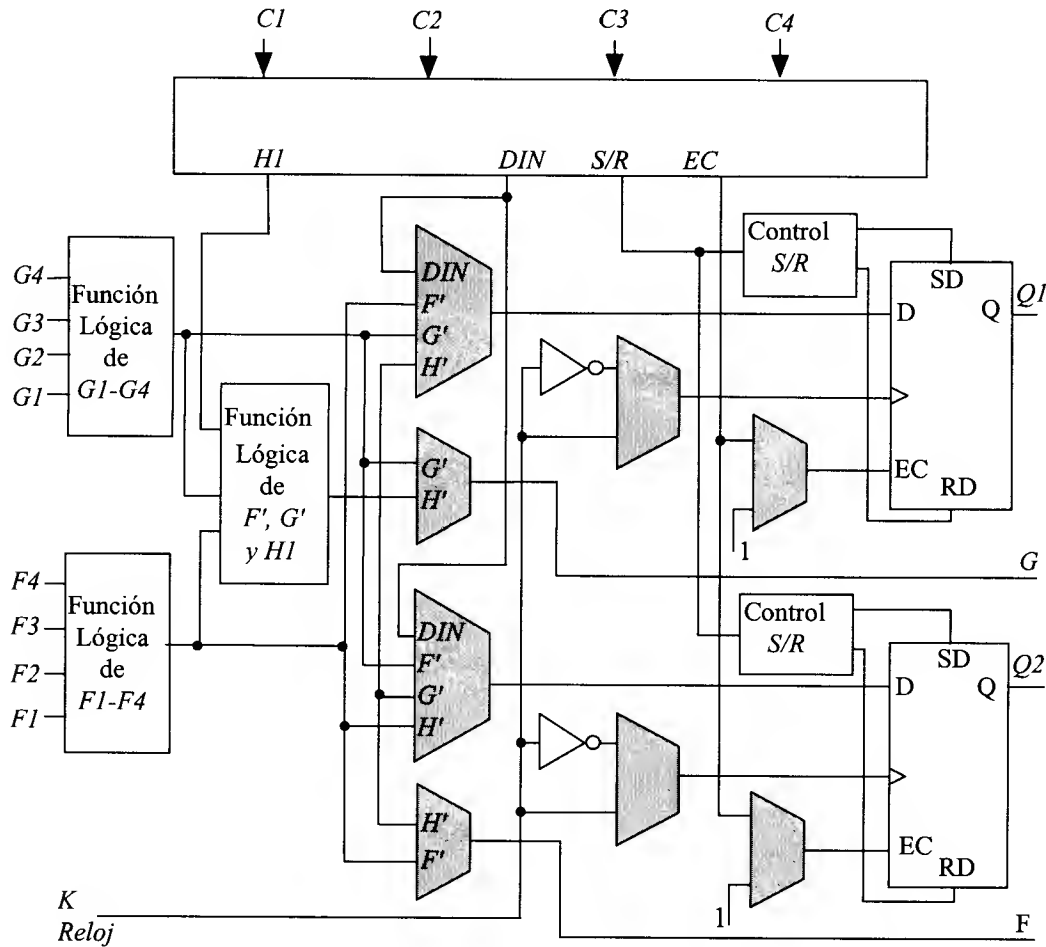
No es sin embargo tan importante el número de puertas o biestables que incluye cada nueva serie como la filosofía de diseño a partir de la configuración por software, y la forma de resolver el problema de las interconexiones. Veamos ahora la estructura interna de los tres elementos básicos (CLB's, IOB's y conexiones).

#### **13.3.1.1. Bloques Lógicos Configurables (CLB's)**

La *figura 13.8* muestra el esquema de la estructura interna de los bloques lógicos configurables de la familia XC-4000. Consta esencialmente de dos biestables ( $Q1$ ,  $Q2$ ) y dos generadores independientes ( $G'$  y  $F'$ ) de funciones lógicas de cuatro variables ( $G4$ ,  $G3$ ,  $G2$ ,  $G1$ ) y ( $F4$ ,  $F3$ ,  $F2$ ,  $F1$ ) que a su vez se pueden combinar para formar una nueva función lógica ( $H'$ ) de tres variables ( $F'$ ,  $G'$  y una variable externa  $H1$ ). Cada CLB posee 13 entradas externas y cuatro salidas.

Las funciones  $G'$ ,  $F'$  y  $H'$  son universales y su programación se realiza leyendo en una memoria LUT el valor del coeficiente correspondiente a cada término mínimo. El término mínimo actúa como dirección de la memoria y su contenido es el bit que determina si tal término mínimo debe o no debe

incluirse en la función. Para implementar estas funciones  $G'$  y  $F'$  de cuatro variables nos hacen falta 16 posiciones de memoria RAM de 1 bit (tantas como términos mínimos). El resto de la memoria RAM incluida en el circuito puede usarse como recurso adicional, cumpliendo las funciones usuales en cualquier otro sistema de cálculo digital.



**Figura 13.8.** Estructura interna del bloque lógico configurable. Los multiplexos (trapezoides) están controlados por el programa de configuración.

Esta forma de programar funciones combinatoriales específicas a partir de una estructura universal y una memoria RAM en la que se almacenan los coeficientes, hace que el retardo de propagación sea independiente de la función concreta que esté operando en una aplicación.

Entre los tres generadores ( $G'$ ,  $F'$  y  $H'$ ), y los biestables D y las salidas existe una red de multiplexores controlados por el programa de configuración que controlan el tráfico de datos y la conectividad funcional efectiva dentro del bloque. Así, las salidas de los generadores pueden abandonar el bloque por dos caminos. Las señales  $F'$  o  $H'$  pueden conectarse a la salida  $F$ . Análogamente,  $G'$  o  $H'$  pueden conectarse a la salida  $G$ . Esto permite un primer modo de configuración del bloque lógico como generador de dos funciones combinacionales independientes de cuatro variables (salidas directas de  $F'$  por  $F$  y de  $G'$  por  $G$ ), como generados de una función de cuatro variables ( $F'$ , por ejemplo) y otra de cinco ( $H'$  de  $G'$  y  $HI$ , por ejemplo) o como generador de un subconjunto de funciones de nueve variables resultado de combinar mediante una función universal de tres variables ( $G'$ ,  $F'$  y  $HI$ ), dos funciones universales de cuatro variables y una variable externa adicional. Cada bloque CLB puede configurarse entonces como un autómata universal de cuatro estados con nueve variables de entrada. Los multiplexores establecen cualquier correspondencia entre las cuatro entradas de control ( $C1$ ,  $C2$ ,  $C3$  y  $C4$ ) y las cuatro señales de control internas ( $HI$ ,  $Din$ ,  $S/R$  y  $EC$ ).

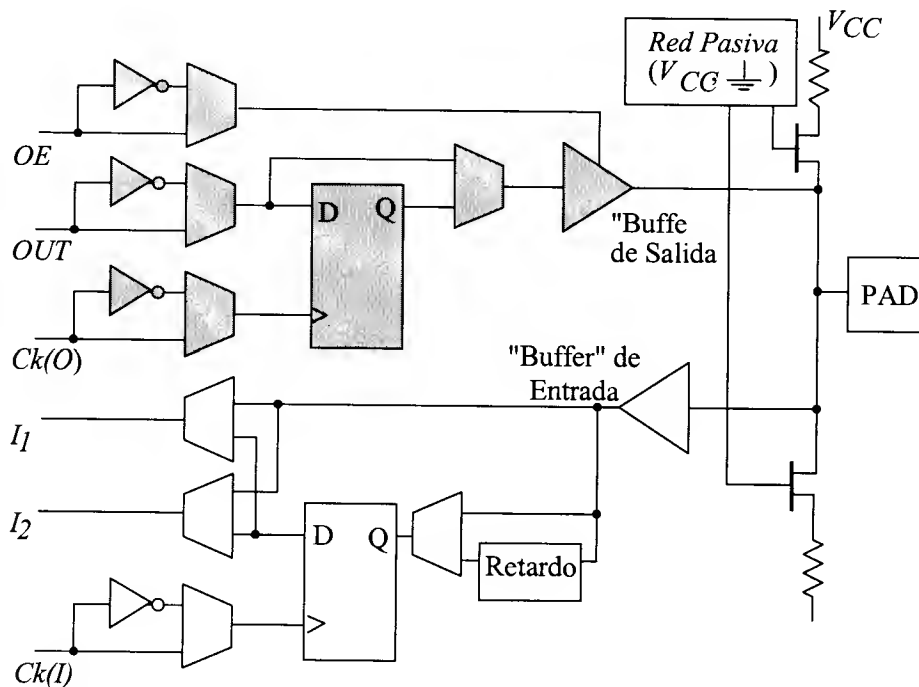
Veamos ahora la parte secuencial del CLB. Está formada por dos biestables D disparados por flancos con un reloj común ( $K$ ) y entradas de facilitación del reloj ( $EC$ ), también comunes. La señal de *set/reset* ( $S/R$ ) puede programarse de forma asíncrona para cada biestable o inhibirse. Existe además una línea de *set/reset* global que no aparece en la *figura 13.8*. Las entradas de los biestables son también programables mediante multiplexos y pueden proceder de  $F'$ ,  $G'$  o  $H'$ , o bien del exterior del bloque a través del terminal de entradas directas,  $DIN$ . Las dos salidas de los biestables ( $Q1$ ,  $Q2$ ), junto con las de los multiplexos de la parte combinacional ( $G$  y  $F$ ), constituyen las cuatro salidas del bloque.

#### **13.3.1.2. Bloques de Entrada-Salida (IOB's)**

Recordemos que en la arquitectura general mostrada en la *figura 13.7*, la matriz de bloques lógicos (CLB's), estaba rodeada a lo largo de todo su perímetro por bloques de entrada-salida (IOB's) igualmente configurables por el usuario. Su misión es actuar de enlace entre los terminales externos del circuito y la matriz de bloques lógicos. Cada IOB controla un terminal y puede configurarse como entrada, como salida o para la transmisión bidireccional de señales. La *figura 13.9* muestra el diagrama de bloques de la estructura interna de estos módulos de entrada salida.



Hay dos líneas de señal,  $I_1$  e  $I_2$ , que pueden actuar como entradas de un biestable D o pasar directamente de acuerdo con el programa de configuración. Hay también, junto a las señales de reloj, una señal de facilitación de la salida ( $OE$ ) que permite la conexión a canales de tres estados (con uno de alta impedancia) o bien su funcionamiento como línea bidireccional. Existen otras funciones también programables, como las resistencias de apoyo a los terminales de la fuente de alimentación, cuya misión es reducir el consumo, conectando a  $V_{CC}$  ó a tierra los terminales que no se usen en un diseño específico.



**Figura 13.10.** Estructura interna de los bloques de entrada/salida que pueden configurarse como entrada, como salida, bidireccionales o en tres estados. Como siempre, los multiplexos y el tipo de salida o entrada deseadas (directa o a través de registro de tres estados) se configura leyendo el contenido de una memoria RAM.

### 13.3.1.3. Conexiones Programables

La clave de todo diseño regular y recursivo y en definitiva de toda teoría modular de autómatas está en la conectividad. Cualquier esquema de conexión que tenga que estar presente en alguna

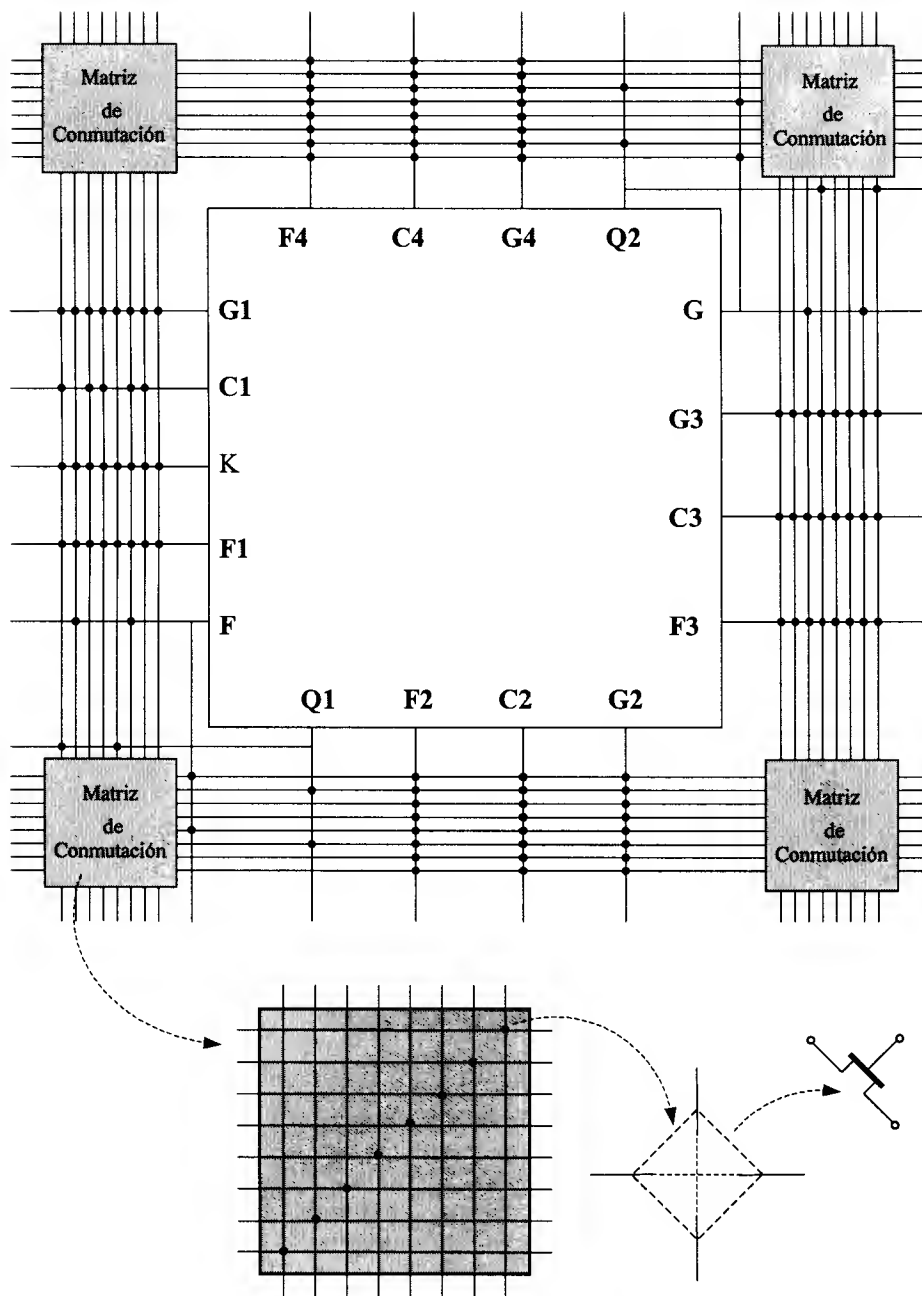
configuración debe de estar considerado en la estructura. Además, debe ser fácil su programación. Evidentemente, la solución de la conectividad total no es razonable ni físicamente realizable para el número de líneas que hay que conectar en un circuito con 900 CLB's. La única alternativa es optimizar la arquitectura estableciendo una jerarquía de conexiones distinguibles por la longitud relativa de su zona de conexión, a partir de una unidad. Tenemos así líneas de longitud básica (entorno de una CLB), líneas de doble longitud (conexión entre CLB's alternadas) y líneas largas (conexiones globales).

Sea cual fuere la longitud del segmento metálico potencialmente conectable, todas las conexiones se realizan a través de matrices de conmutación programables, tal como se ilustra en la *figura 13.11*. El número concreto de líneas que se muestran en el dibujo no es significativo ya que en la realidad depende del número de bloques. Al aumentar el tamaño de la matriz de CLB's, aumenta proporcionalmente el número de líneas metalizadas usadas para su interconexión.

La *figura 13.10.a* muestra el esquema de conexiones para las líneas que rodean a un CLB. Es decir, es una conectividad de longitud básica que sólo permite la conexión de un módulo con sus vecinos. Sin embargo, programando adecuadamente las matrices de conmutación se puede extender la conectividad.

Cada matriz de conmutación, tal como se ilustra en la *figura 13.10.b*, consta de seis transistores de paso que pueden programarse para establecer la conectividad funcional entre cada par de terminales de los que entran al nudo permitiendo así prolongar la ruta de datos (atravesando el nudo), girar a la izquierda o girar a la derecha o seguir a la vez cualquier combinación de estos caminos, hasta llegar al siguiente nudo de forma que cada CLB puede conectarse con cualquier otro a través del camino programado sobre la malla de matrices de conmutación.

Una segunda posibilidad de conexión es a través de las líneas de doble longitud que sólo entran a sus matrices de conmutación tras pasar por el lado de dos CLB's. Su función es conectar de forma eficiente a las distancias intermedias. Finalmente, las líneas de conexión global forman un entramado de segmentos metálicos interconectados que cubren toda la matriz de CLB's y realizan funciones de distribución de las señales de control y los pulsos del reloj, junto con la posibilidad de ciertas conexiones en las que el tiempo es crítico. Su conexión con los CLB's es a través de amplificadores de tres estados o a través de las líneas adyacentes y en ambos casos es programable.



**Figura 13.11.** Conexiones de una CBL a sus líneas adyacentes de longitud básica. **(a)** Esquema general. **(b)** Ilustración de la matriz de conmutación sintetizada mediante seis transistores de paso que controlan la conexión funcional entre cada una de las líneas aferentes al nodo.

De hecho, todos los módulos (CLB, IOB y conexiones) son re-configurables dinámicamente. Esto permite evaluar y refinar los diseños mientras estos operan en su entorno real, realizar distintas funciones en intervalos de tiempo diferentes, incluir programas de autodiagnóstico y admitir la reprogramación cuando cambia el entorno, abriendo las puertas al diseño digital adaptivo y con posibilidades de incorporar procesos de aprendizaje.

### **13.3.2. Familias Virtex-EM y ORCA-4**

La última familia introducida por XILINX es la Virtex-EM con densidades de hasta 400.000 puertas lógicas ó 32.000 celdas lógicas, con posibilidad de integración en sistemas digitales con frecuencias de reloj superior a los 100 MHz, con mayores facilidades de interconexión y con tres niveles de memoria RAM: la que lleva incorporada los CLBs, la de los bloques “empotrados”, y la posibilidad de acceso a un tercer nivel de memoria fuera del circuito. También puede incorporar núcleos IP (“Intellectual Property cores”) con funciones complejas prediseñadas para realizar tareas específicas tal como ya vimos al estudiar las ultimas series de CPLDs y tal como se usa también en los circuitos integrados de aplicaciones específicas (ASIC). De nuevo los diseños de circuitos FPGA, inicialmente de arquitectura simétrica y grano medio, incorporan elementos arquitecturales y funcionales idénticos a los usados por los CPLDs (bloques de memoria RAM y núcleos IP), dejando borrosa la frontera.

Una evolución análoga han seguido los FPGAs de Lucent Technologies, ORCA (Optimized Reconfigursble Cell Arrays), que también son arquitecturas modulares de bloques lógicos (llamados aquí PLC, Programable Logic Cells), rodeadas de celdas programables de entrada y salida (PIC) y múltiples facilidades de interconexión con reconfigurabilidad basada en el uso de memoria SRAM.

## **13.4. ¿DÓNDE TERMINA LA ELECTRÓNICA Y DÓNDE EMPIEZA LA PROGRAMACIÓN?**

Hasta la década de los setenta parecía que estaba clara la distinción entre electrónica y programación porque la Electrónica Digital evolucionaba por dos caminos:

- a) Desarrollo de circuitos de propósito general.*
- b) Desarrollo de circuitos de aplicaciones específicas.*

En el camino de los circuitos de propósito general la meta era conseguir sistemas procesadores en arquitectura von Neumann cada vez más rápidos y potentes, dando por supuesto que el propósito de la Electrónica Digital y la arquitectura de computadores era desarrollar hardware general y eficiente, independientemente de su posterior programación. Para pasar de una aplicación a otra cambiamos el programa pero el hardware permanece. Sus módulos combinacionales y secuenciales son los mismos y la interconectividad (la arquitectura) es fija. El hardware no se reconfigura. La lógica no se reprograma.

La otra vía de evolución se orienta “a la demanda”. Un usuario necesita un circuito que le resuelva un determinado problema de cálculo, comunicación, instrumentación o control y los electrónicos se lo diseñan. Bien entendido que sólo le va a servir para esa aplicación. Son circuitos de “propósito especial” o de “aplicaciones específicas” que hacen un uso muy eficiente del semiconductor, son rápidos y permiten privacidad en los desarrollos. Sin embargo, la posibilidad de programación es muy baja y la conectividad y las funciones locales son fijas. Aquí el diseño de una aplicación no sale de las fronteras de la electrónica.

Sin embargo, en la década de los setenta comienza el desarrollo de la lógica programable desde las modestas PROM en las que la función se podía programar sólo una vez. Así, a partir de un mismo tipo de circuito, el usuario podía obtener su diseño particular. El siguiente paso fue la posibilidad de programar y reprogramar eléctricamente un número prácticamente ilimitado de veces (EEPROM y EEPROM-flash) y el uso de celdas SRAM en tecnología CMOS para almacenar la configuración del sistema. Como el contenido de estas celdas puede modificarse por programa y, a su vez, puede modificar la conectividad y la función de los bloques lógicos, empiezan a mezclarse electrónica y programación. El incremento gradual de la complejidad de la lógica programable en densidad, velocidad y funcionalidades específicas acentúa más la borrosidad de esta frontera.

La mezcla entre electrónica y programación es todavía más evidente en los nuevos sistemas dinámicamente reconfigurables que reciben nombres tales como *soft-hardware*, *hardware virtual* o *caché* y *hardware adaptable y evolutivo*. En todos los casos, aunque con matices diferentes, nos estamos refiriendo a circuitos electrónicos cuya función puede cambiarse fácilmente en tiempo de ejecución, durante la operación normal del sistema.

Pensemos por ejemplo en un conjunto de aplicaciones en las que sabemos que si dispusiéramos de un hardware específico para cada una de ellas el software sería más eficiente aumentando la

velocidad de cálculo. Podemos entonces desarrollar previamente un conjunto de *programas de configuración* del hardware de forma que para cada una de las aplicaciones de ese conjunto se descargue sobre la RAM del circuito digital reconfigurable la configuración que mejor se adapta a las exigencias computacionales de esa aplicación. Decimos entonces que la arquitectura del sistema se ha adaptado para optimizar la solución de esa aplicación específica.

Lo que en el fondo hemos hecho ha sido usar la capacidad de reconfiguración de las FPGAs para implementar una parte del algoritmo que ahora se reparte entre la electrónica y la programación convencional.

Esta posibilidad de especializar la electrónica de un computador para hacer más eficiente la ejecución de los programas correspondientes a una determinada aplicación abre las puertas al concepto de *aprendizaje* en un sentido no trivial. La posibilidad de que un circuito digital “*aprenda*” o “*se adapte*” es una consecuencia del reparto en cada momento de la carga computacional entre la operación efectiva de una lógica estática y el cálculo de la nueva configuración de ese circuito. Es decir, constantemente hay una parte de la operación de un circuito digital que se dedica a calcular y hacer efectiva su propia *reconfiguración* con la que volverá a operar en el siguiente intervalo de tiempo.

Finalmente, el propio procedimiento de diseño se hace también híbrido. Hasta un cierto nivel de complejidad diseñábamos siguiendo al detalle los procesos intermedios (niveles de puertas, registros, funciones de excitación de los biestables, temporizadores, relojes, contadores, unidades de control, etc...). La llegada de la lógica programable introduce una arquitectura general muy próxima al modelo matemático de autómatas (matrices AND/OR para la lógica combinacional y ristas de biestables D en las macroceldas de salida, para la lógica secuencial) y descansa en gran medida en un entorno de programación que pasa de la especificación funcional a la configuración final del CPLD o FPGA que la satisface por caminos que permanecen esencialmente ocultos al diseñador.



## REFERENCIAS BIBLIOGRÁFICAS

1. Altera Corporation. FLEX 10K. “*Embedded Programmable Logic Family*”. Data Sheet A.DS.F10K-04.02. Mayo, 2000.
2. Hennie, F.C. “*Iterative Arrays of Logical Circuits*” MIT Press, 1961
3. Lucent Technology: ORCA Series 4. *Field Programmable Gate Arrays*. Preliminary Data Sheet. August 2000.
4. Sharma, A.K.: “*Programmable Logic Handbook: PLDs, CPLDs & FPGAs*”. McGraw-Hill, 1998.
5. XILINX. “*The Programmable Gate Array Data Book*”. San José, CA, USA, 1992
6. XILINX. “*XC4000E and XC4000X Series Field Programmable Gate Arrays*”. Product Specification. May, 1999.





---

## ***Resumen de Física de Semiconductores***

---

La práctica totalidad de los dispositivos electrónico actuales está realizada sobre materiales semiconductores. Un semiconductor es un cristal del grupo IV del sistema periódico (Silicio, Germanio) o un compuesto III-V (Arseniuro de Galio) que se caracteriza por tener una conductividad, distinta de la de los metales y la de los aislantes, y que depende fuertemente de dos parámetros: (a) la temperatura y (b) el grado de impurificación.

En el cero absoluto un semiconductor puro, al que llamamos *intrínseco*, se comporta como un aislante y no conduce. Sin embargo, al aumentar la temperatura por encima de un cierto valor umbral empieza a conducir. La segunda propiedad es también muy interesante. La conductividad de un semiconductor puro (intrínseco) se puede *controlar* con gran facilidad mediante su impurificación con pequeñísimas dosis de átomos de otro material de los grupos III y V convirtiéndolo en semiconductor extrínseco de tipo P o de tipo N. En este resumen vamos a contestar a las siguientes cuestiones:

1. ¿Por qué se comportan así los semiconductores?: Teoría de Bandas.
2. ¿Cómo se caracterizan las situaciones de equilibrio en P y N?.

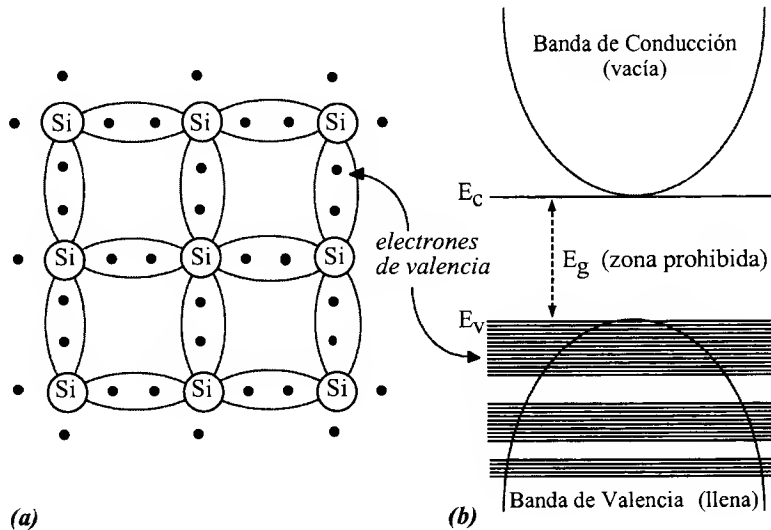


### 3. ¿Cómo se transporta y recombina la carga en exceso?

Al contestar a estas tres cuestiones obtenemos las bases de Física de Semiconductores necesarias para comprender el funcionamiento de los dispositivos electrónicos, diodos y transistores.

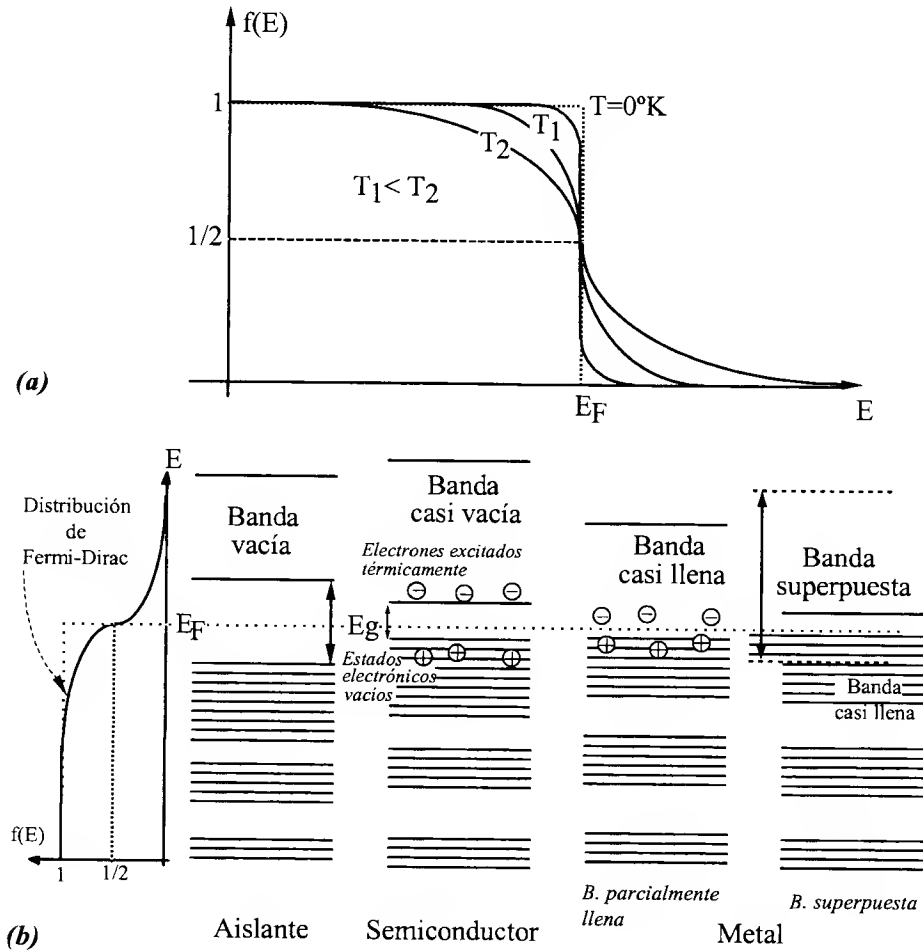
## A.1. TEORÍA DE BANDAS

Un cristal de silicio, semiconductor que usaremos en adelante como referencia, posee una estructura periódica en forma de red con cuatro electrones de valencia en la última capa, tal como se ilustra de forma bidimensional en la *figura A.1.a*. Como consecuencia de esta periodicidad en el cristal las energías que pueden ocupar sus electrones se distribuyen en bandas de forma que hay desdoblamiento de los niveles individuales de energía dando lugar a intervalos de energías permitidas separados por otros intervalos de energías prohibidas. De todas las bandas sólo son importantes las dos últimas llamadas banda de conducción, con energía  $E_C$  y generalmente vacía y banda de valencia, con energía  $E_V$  y generalmente llena (*figura A.1.b*).



**Figura A.1.** Efecto de la periodicidad en un cristal semiconductor. **(a)** En el cristal hay una estructura cúbica con una celda única que se repite. **(b)** En el modelo de bandas aparecen regiones de energías permitidas alternándose con otros de energías prohibidas. Las dos últimas bandas son la de Valencia ( $E_V$ ) y la de Conducción ( $E_C$ ).

Una primera aplicación del modelo de bandas es la distinción entre metales, aislante y semiconductor. Para compararlos usamos la *figura A.2.b* en la que hemos superpuesto al diagrama de bandas la función de distribución de Fermi-Dirac,  $f(E)$ , que representa la probabilidad de ocupación por un electrón de un cierto estado de energía  $E$ . Obsérvese que hasta la energía del nivel de Fermi,  $E_F$ , todos los estado están ocupados,  $f(E) = 1$  para todo  $E \leq E_F$  en el cero absoluto. Al aumentar la temperatura la función  $f(E)$  se suaviza en torno a  $E_F$ , tal como se ilustra en la parte (a) de la *figura A.2*.



**Figura A.2.** Distinción entre metales, semiconductores y aislantes. **(a)** Función de distribución de Fermi-Dirac. **(b)** Estructura de bandas en la que se ha superpuesto la función de Fermi-Dirac para observar las bandas que están total o parcialmente llenas o vacías al variar la temperatura.

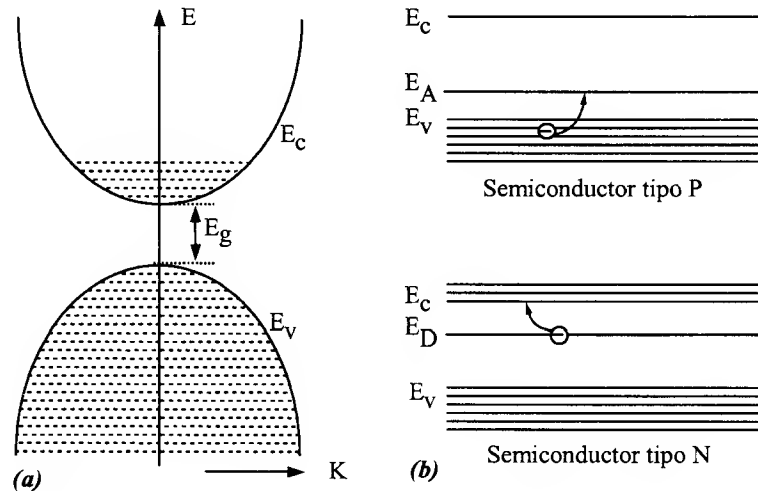
En un aislante el número de electrones en el cristal es justo suficiente para llenar completamente un determinado número de bandas, empezando por la baja de energía. Después hay una región de energías prohibidas y a partir de aquí todas las bandas están completamente vacías. Así, no es posible la conducción. No hay corriente de electrones libres. Una banda completamente llena no contribuye a la conducción pues "no tienen a donde ir" los electrones. Una banda completamente vacía tampoco contribuye a la conducción, claramente. La tercera posibilidad que sería el paso de portadores de una banda a la siguiente como consecuencia de la agitación térmica es poco probable, dado el gran valor del intervalo de energías prohibidas "gap" y de la posición del nivel de Fermi.

Para un semiconductor, el "gap" entre las bandas totalmente llena y vacía es mucho menor y al aumentar la temperatura existe una probabilidad apreciable de que la excitación térmica produzca electrones libres. Es decir, que se vacíe un poco la banda de valencia y pasen electrones a estados energéticos de la banda de conducción. Quedan así dos bandas parcialmente llenas y por consiguiente, la conducción es posible. Para un metal hay dos posibles estructuras de bandas, la primera corresponde a una banda parcialmente llena y la segunda a un caso de "solape". Vemos que aún a la temperatura de cero absoluto puede haber conducción.

En los semiconductores, la conductividad depende fuertemente de la temperatura. Así, en el cero absoluto son aislantes. Al aumentar la temperatura aparece una conductividad finita, debida a los portadores generados térmicamente, pero mucho menor que la de los metales. El aspecto realmente importante del comportamiento de un semiconductor (en cuanto a la posibilidad de realizar distintos dispositivos electrónicos) es la facilidad con la que se puede controlar su conductividad controlando el nivel y tipo de "impurezas" que se le añaden. Estas impurezas son tan pocas que apenas modifican las estructuras física y energética del semiconductor. Su estructura de bandas se conserva. Sin embargo, los estados electrónicos correspondientes a los electrones en exceso o defecto de estas impurezas serán localizados y toman valores situados en el interior de la banda de energías prohibidas. Por eso modifican tan ostensiblemente la dinámica del semiconductor.

En un semiconductor intrínseco a temperatura ambiente todas las bandas hasta la de valencia,  $E_v$ , están completamente llenas y por consiguiente no contribuyen a la conducción. A partir de la banda de conducción,  $E_c$ , todas las regiones de energías permitidas están completamente vacías y tampoco contribuyen a la conducción. Así pues, para estudiar el transporte controlado de carga en los dispositivos semiconductores sólo nos interesa considerar las bandas de conducción y de valencia donde como consecuencia de modificaciones de la temperatura en los semiconductores intrínsecos y

como consecuencia de procesos de impurificación en los extrínsecos, se modifica el estado de ocupación de bandas parcialmente llenas, dando lugar a procesos de conducción. El intervalo de energías prohibidas que separa a las bandas de conducción y valencia,  $E_g$ , es crucial por dos razones. Su valor mide el salto energético necesario y además, en él se sitúan los estados asociados a las impurezas. La *figura A.3.a* muestra la forma aproximada del tope de la banda de valencia y del fondo de la banda de conducción. Esta forma suele ser parabólica, además la inferior suele estar casi llena, con pocos estados desocupados y la superior, casi vacía, con muy pocos estados electrónicos ocupados. Así están las cosas en los semiconductores intrínsecos, no se altera apenas la estructura periódica de la red cristalina por lo que sigue siendo válida la representación en bandas.

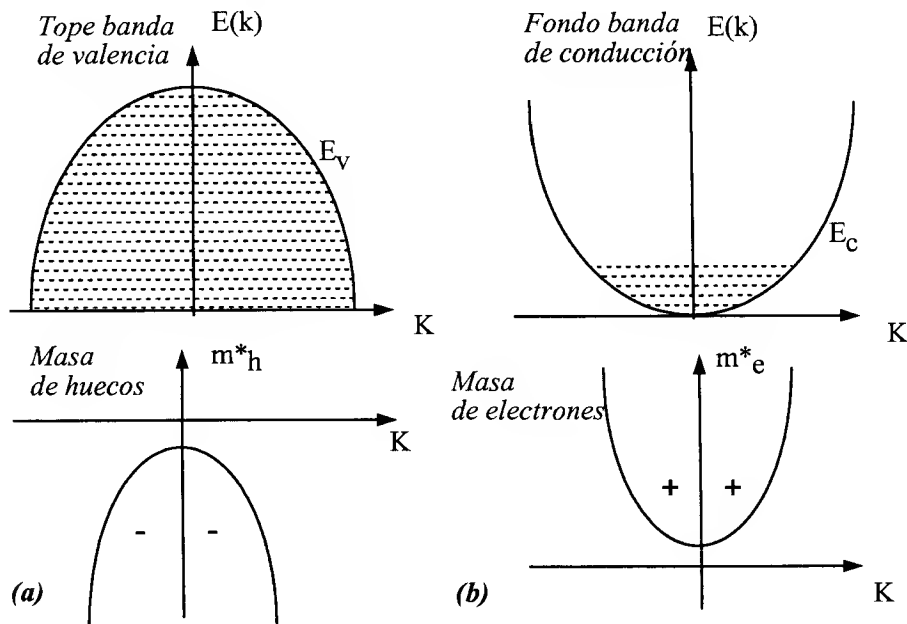


**Figura A.3.** (a) Forma parabólica aproximada del tope de Banda de Valencia y del fondo de la Banda de Conducción donde se desarrolla la conducción en semiconductores. (b) Sustitución de las parábolas por rectas tangentes en  $K=0$  e ilustración del efecto de la impurificación que produce niveles "aceptores",  $E_A$ , cerca de  $E_v$  en los semiconductores extrínsecos tipo P y niveles "dadores",  $E_D$ , cerca de  $E_c$  en los tipo N.

Las bandas no se alteran, sin embargo los pocos átomos que se entremezclan en los intersticios de la red hacen que en el modelo de la teoría de bandas aparezcan estados ligados con energías localizadas en el intervalo de energías prohibidas (entre  $E_v$  y  $E_c$ ), dando lugar a semiconductores extrínsecos tipo P (si los estados de las impurezas están cerca del tope de  $E_v$ ) o tipo N (si están cerca del fondo de  $E_c$ ). Estos estados están prácticamente ionizados todos a temperatura ambiente y dominan la conductividad dando lugar a toda la tecnología bipolar de producción de dispositivos

electrónicos, basada en la unión de un semiconductor intrínseco tipo P (dominado por conducción en banda de valencia) con un semiconductor extrínseco tipo N con conducción dominante en banda de conducción (unión P-N).

Para estudiar el movimiento de carga en un semiconductor como consecuencia de la aplicación de un campo externo necesitamos introducir los conceptos de "masa efectiva",  $m^*$  y "hueco",  $h$ . La masa efectiva incluye el efecto del potencial periódico de la red cristalina de forma que podemos estudiar el movimiento de un electrón por una banda suponiendo que no existe el potencial de la red pero que la masa del electrón no es su masa gravitatoria,  $m_e$ , sino su masa efectiva,  $m^*_e$ . Esta masa resume el comportamiento mecano-cuántico del electrón y por consiguiente depende de la forma de las bandas y es algo "rarilla" ya que puede ser positiva o negativa (*figura A.4*). En el tope de la banda de valencia (*figura A.4.a*) es negativa indicando que en esa región los electrones se comportan como cargas positivas, dando lugar al concepto de hueco del que hablaremos en seguida. En el fondo de la banda de conducción (*figura A.4.b*)  $m^*$  es positiva, dando lugar al comportamiento esperado en los electrones.



**Figura A.4.** Representación de las masas efectivas en el tope de la banda de valencia (a) y en el fondo de la banda de conducción (b).

Los estados electrónicos asociados al tope de la banda de valencia (que está casi llena) tienen asociados masas efectivas negativas, de forma que es aconsejable cambiar la representación y en vez de estudiar el movimiento de los muchos estados ocupados, estudiar el correspondiente a los pocos estados no ocupados. Decimos entonces que la conducción en la banda de valencia se realiza por huecos y en la banda de conducción, por electrones. Es importante señalar que un hueco está asociado a un estado electrónico no ocupado por un electrón, pero no debemos confundirlo con una vacante o un estado iónico en el cristal real. Dicho de otro modo, el hueco está en las bandas, no en el cristal en el que, obviamente, sólo existen electrones.

## A.2. ¿CÓMO SE CARACTERIZAN LAS SITUACIONES DE EQUILIBRIO?.

En situaciones de equilibrio, a temperatura constante y sin presencia de campos externos, eléctricos o magnéticos, un semiconductor se caracteriza por una concentración de electrones,  $n$ , en la banda de conducción y una concentración de huecos,  $p$ , en la banda de valencia, siendo

$$n = n(T, E_F, m_e^*) = N_c \cdot e^{-(E_C - E_F)/KT}$$

$$p = p(T, E_F, m_h^*) = N_v \cdot e^{-(E_F - E_V)/KT}$$

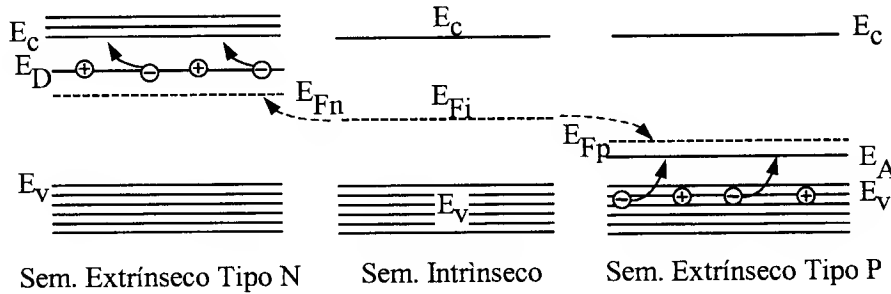
En un semiconductor intrínseco, los electrones y los huecos se producen exclusivamente mediante excitación térmica. Estos portadores se producen a pares por lo que ambas concentraciones deben ser iguales y sólo función de la temperatura.

$$n_0 = p_0 = n_i(T)$$

Hemos visto la situación en semiconductores "puros" (intrínsecos). Veamos que pasa con los semiconductores extrínsecos P y N.

Si impurificamos un semiconductor intrínseco con una cierta densidad de átomos dadores ( $N_D$ ) o aceptores ( $N_A$ ) obtenemos semiconductores extrínsecos tipo N ó P. Desde el punto de vista de la teoría de bandas podemos suponer que estas densidades son lo suficientemente bajas como para no perturbar la periodicidad del cristal y las podemos representar añadiendo una serie de *nuevos estados energéticos localizados* con energías  $E_D$  y  $E_A$  en el intervalo de energías prohibidas. Esto despliega la

posición del nivel de Fermi de forma que ahora en los semiconductores tipo N obtenemos un nivel  $E_{Fn}$  próximo a  $E_C$  y en los P un nivel diferente  $E_{Fp}$ , próximo a  $E_V$ , tal como ilustra la *figura A.5*.



**Figura A.5.** Paso de un semiconductor intrínseco a los extrínsecos P y N con el correspondiente desplazamiento del nivel de Fermi.

Es decir, el efecto de las impurezas, tanto dadoras como aceptoras, ha sido desplazar el nivel de Fermi pero podemos mantener las expresiones propuestas anteriormente para el cálculo de las concentraciones,  $n$  y  $p$  usando dos niveles de Fermi:

$$\text{Semiconductor tipo N} \Rightarrow E_F = E_{Fn}$$

$$\text{Semiconductor tipo P} \Rightarrow E_F = E_{Fp}$$

Hay que hablar ahora de portadores  $n$  en un semiconductor N,  $n_n$ , portadores  $p$  en N,  $p_n$  y análogamente portadores  $n$  y  $p$  en un semiconductor P,  $n_p$  y  $p_p$ .

$$n_n = N_c e^{-(E_C - E_{Fn})/KT}, p_n = N_v e^{-(E_{Fn} - E_V)/KT}$$

$$p_p = N_v e^{-(E_{Fp} - E_V)/KT}, n_p = N_c e^{-(E_C - E_{Fp})/KT}$$

Lo que diferencia a los semiconductores N de los P es la posición de  $E_F$ . Así, en el tipo N:

$$(E_C - E_{Fn}) \ll (E_{Fn} - E_V) \Rightarrow p_n \ll n_n$$

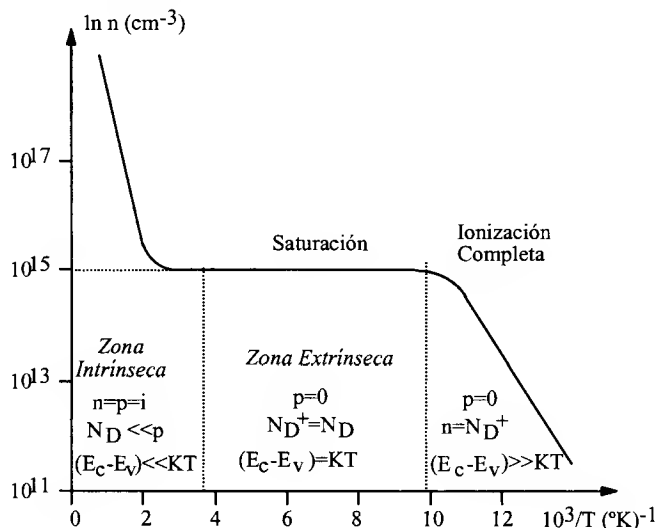
y en los semiconductores tipo P:

$$(E_{Fp} - E_V) \ll (E_C - E_{Fp}) \Rightarrow n_p \ll p_p$$

mientras que en los intrínsecos ambas concentraciones eran iguales y sólo dependían de la temperatura:

$$n = p = n_i(T)$$

En un semiconductor extrínseco, las concentraciones están dominadas por las impurezas. Veamos como se llega a esta situación aproximada al aumentar la temperatura de un semiconductor a través de la representación gráfica de la densidad de portadores en función de la temperatura para una muestra de silicio tipo N con  $N_D = 10^{15} \text{ cm}^{-3}$  (figura A.6).



**Figura A.6** Variación de la carga,  $n$ , para un semiconductor extrínseco tipo N al aumentar la temperatura. El eje de abscisas representa el inverso de la temperatura y el de ordenadas el logaritmo de la concentración. En la zona intrínseca la pendiente es  $(E_C - E_V)/2$ . Es decir, el "gap" de energías prohibidas. Antes de saturación, en la zona extrínseca la pendiente es  $(E_C - E_D)/2$ , mucho menor que  $(E_C - E_V)/2$ .

En esta curva de concentración distinguimos tres zonas: *intrínseca*, de *agotamiento* ó *saturación* y de *ionización incompleta*. Como en el eje de abscisas representamos  $10^3/T$ , al aumentar la temperatura nos desplazamos de derecha a izquierda y en este sentido describiremos la curva.

Para bajas temperaturas es muy difícil que un electrón salte de la banda de valencia a la banda de conducción y los pocos electrones que hay en la banda de conducción proceden de la ionización de impurezas dadoras y el número de huecos es pequeño. Supongamos  $p=0$ . La conducción está



dominada por impurezas. Al aumentar la temperatura rápidamente se ionizan todas las impurezas,  $N_D^+ = N_D$ , se agotan y llegamos a una zona de saturación correspondiente a un intervalo de temperatura en las que la concentración es constante. Finalmente, si seguimos aumentando la temperatura, llega una zona en la que la densidad de electrones en banda de valencia debido a la ionización de impurezas es mucho menor que la debida a los pares electrón-huecos. Estamos en el rango de *comportamiento intrínseco*.

### A.3. ¿CÓMO SE TRANSPORTA Y RECOMBINA EL EXCESO DE CONCENTRACIÓN?.

Hasta ahora hemos considerado situaciones de equilibrio en las que todo mecanismo de generación de pares electrón-hueco tenía asociado otro de recombinación a igual velocidad consiguiendo así situaciones estacionarias caracterizadas por un único nivel de Fermi y con distribuciones  $n$  y  $p$  que cumplían:

$$n = N_c \cdot e^{(E_F - E_c)/KT}$$

$$p = N_v \cdot e^{(E_v - E_F)/KT}$$

$$n \cdot p = n_i^2(T)$$

Sin embargo si hay un aporte de energía externa (iluminación del material o inyección de portadores, por ejemplo) deja de cumplirse el principio de Balance Detallado y aparece un exceso de portadores en el semiconductor de forma que

$$n \cdot p > n_i^2 \Rightarrow \text{inyección}$$

$$n \cdot p < n_i^2 \Rightarrow \text{extracción}$$

A partir de aquí buscamos la caracterización de la vuelta al equilibrio (procesos de recombinación) y el transporte del exceso de portadores como consecuencia de la aplicación de campos externos (corrientes de arrastre) o de la difusión asociada a todo gradiente de concentración.

En metales es prácticamente imposible alterar la concentración de portadores libres en el volumen del material. Sin embargo en semiconductores esto es sencillo debido a la existencia de dos

tipos de portadores que permiten alterar sus concentraciones individuales ( $n \neq n_0 + \Delta n$ ,  $p \neq p_0 + \Delta p$ ) sin alterar apenas la neutralidad eléctrica global ( $n = p$ ), y alterando fuertemente la conductividad. Esta es la base de la mayoría de las estructuras físicas soporte de los dispositivos electrónicos. En este apartado vamos a estudiar el *transporte de estos portadores en exceso* y después comentaremos los *mecanismos de recombinación*.

El estudio del transporte lo planteamos en cuatro puntos:

**A.3.1.** *Corriente de arrastre ( $J_{na}$ ,  $J_{pa}$ ).*

**A.3.2.** *Corriente de difusión ( $J_{nd}$ ,  $J_{pd}$ ).*

**A.3.3.** *Ecuación de continuidad.*

**A.3.4.** Solución de un *ejemplo* de interés para el estudio del dispositivo electrónico.

Finalmente, en A.3.5 comentaremos los procesos de recombinación.

### **A.3.1. Arrastre.**

Si aplicamos un campo externo o tenemos un gradiente de concentración aparece una aceleración de los electrones en la banda de conducción, compensada por choques con la red o con impurezas dando lugar a un valor medio en la corriente. Como en el semiconductor existen dos bandas parcialmente llenas la corriente total poseerá cuatro componentes, dos de arrastre para huecos y electrones y dos de difusión. Así,

$$\overline{J_{total}} = \overline{J_n} + \overline{J_p} \quad \text{siendo} \quad \begin{cases} \overline{J_n} = \overline{J_{na}} + \overline{J_{nd}} \\ \overline{J_p} = \overline{J_{pa}} + \overline{J_{pd}} \end{cases}$$

Consideremos primero el problema de la conducción en una banda debida a un campo externo. Tenemos entonces, para la banda de conducción por ejemplo, una expresión de la forma

$$\overline{J_{na}} = \sigma \cdot \overline{E} = q \cdot \mu_n \cdot n \cdot \overline{E}$$

donde  $\sigma$  es la conductividad,  $\mu$  y  $n$  la movilidad y la densidad de portadores respectivamente y  $\overline{E}$  el campo eléctrico, de acuerdo con el tratamiento clásico en metales. Aquí, en semiconductores tenemos que repetir el proceso para la otra banda,

$$\overline{J_{pa}} = \sigma \cdot \overline{E} = q \cdot \mu_p \cdot p \cdot \overline{E}$$

Al aplicar un campo externo aparece una componente de arrastre que se superpone a la velocidad propia de la agitación térmica y que da lugar a un desplazamiento neto de carga en la dirección del campo y por consiguiente a una densidad neta de corriente.

### A.3..2. Difusión

Hemos visto la componente de corriente debida a un campo eléctrico que en un semiconductor posee a su vez dos términos, uno para el movimiento de electrones en la banda de conducción y otro para el movimiento de huecos en la banda de valencia. La otra componente se debe a los fenómenos de difusión propios de toda situación en la que existe un gradiente de concentración y de distinto signo para huecos que para electrones:

$$J_p = -|q| \cdot D_p \cdot \nabla p$$

$$J_n = |q| \cdot D_n \cdot \nabla n$$

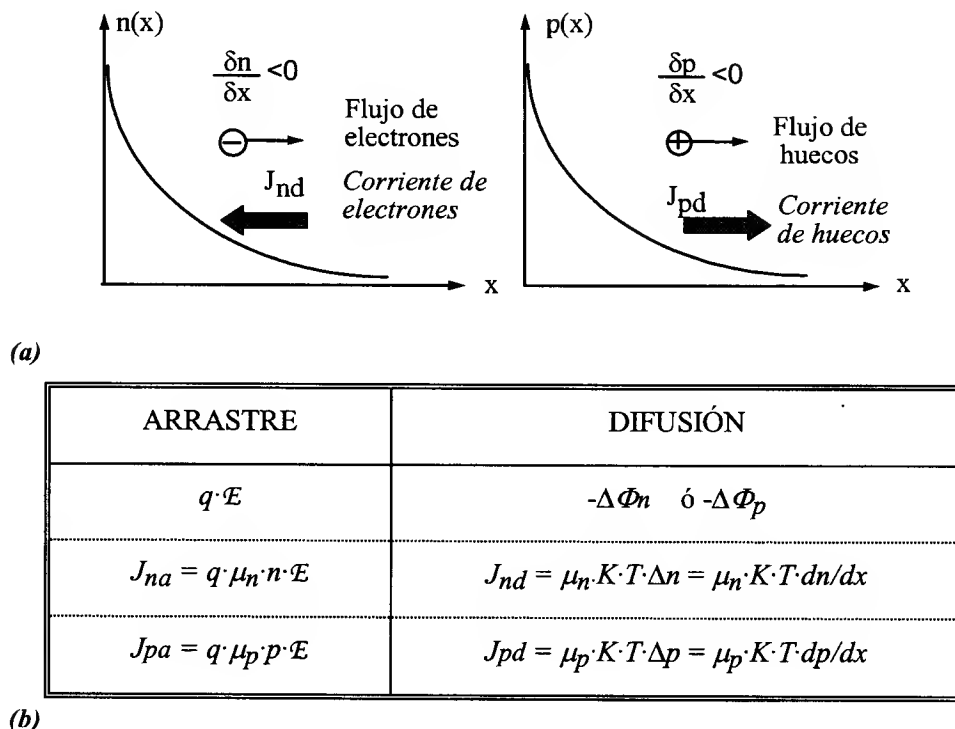
donde  $D_p$  y  $D_n$  son los coeficientes de difusión para huecos y electrones respectivamente y están asociados con la movilidad de acuerdo con las relaciones de Einstein.

$$D_n = \frac{\mu_n KT}{|q|}, \quad D_p = \frac{\mu_p KT}{|q|}$$

Estas relaciones se obtienen a través de la solución de la ecuación de Boltzmann, o bien analizando un caso particular de corriente total nula en el que la difusión cancele el arrastre. La *figura A.7.a* muestra la diferencia entre el flujo de portadores y el sentido de la corriente asociada. Obsérvese que para el caso de los huecos una situación de gradiente negativo ( $p$  decrece al aumentar  $x$ ) el flujo de portadores va hacia la derecha dando lugar a una densidad de corriente asociada  $J_{pd} = -q \cdot D_p \cdot p$  del mismo sentido. En cambio para un gradiente de electrones del mismo tipo y signo el flujo de electrones hacia la derecha lleva asociada una densidad de corriente  $J_{nd} = q \cdot D_n \cdot n$  de sentido opuesto, debido al signo opuesto de ambas cargas.

La expresión de la componente de difusión se puede obtener a partir de la de arrastre sustituyendo la expresión de la fuerza en aquel caso ( $q \cdot E$ ) por la expresión correspondiente cuando

existe un gradiente de concentración consecuencia, por ejemplo, de una inyección de portadores. Tenemos así la tabla resumen de la *figura A.7.b*.



**Figura A.7.** (a) Corrientes de difusión de huecos y electrones debidas a la existencia de un gradiente de concentración. (b) Tabla resumen de las corrientes de arrastre y difusión para huecos y electrones.

### A.3.3. Ecuaciones De Continuidad

Una vez conocidas las densidades de corriente vamos a obtener la ecuación de continuidad que rige el transporte de carga en situaciones tanto transitorias (vuelta al equilibrio, por ejemplo) como estacionarias. La ecuación de continuidad nos relaciona la variación espacial de la densidad de corriente ( $J$ ) con la variación temporal de la densidad de carga ( $\rho$ ):

$$\frac{\partial \rho}{\partial t} = - \frac{\partial J}{\partial x}$$

Esta es la conocida *ecuación de difusión clásica*. Sin embargo, para aplicarla al estudio de los semiconductores debemos ampliarla por dos razones:

1. La conducción se realiza en dos bandas (hay dos tipos de portadores  $p$  y  $n$ ).
2. Existen mecanismos de generación y recombinación.

Con esto obtenemos las siguientes ecuaciones

$$\frac{\partial n}{\partial t} = G_n - R_n + \frac{1}{q} \frac{\partial J_n}{\partial x}$$

$$\frac{\partial p}{\partial t} = G_p - R_p - \frac{1}{q} \frac{\partial J_p}{\partial x}$$

que afortunadamente se simplifican para los casos de interés como vamos a ver a continuación.

### A.3.4. Difusión de Portadores Inyectados

Supongamos que por iluminación o a través de una unión P-N por ejemplo, generamos una fuente constante de portadores en exceso inyectados en  $x=0$  (figura A.8). Se genera entonces un proceso de difusión hacia el interior del semiconductor que en estado estacionario da lugar a un gradiente de concentración  $p_1(x)$ .

Para obtener la forma de ese gradiente resolvemos la ecuación de continuidad simplificada ( $G-R=0$  y  $\frac{\partial p}{\partial t}=0$ ). Queda entonces.

$$D_p \frac{\partial^2 p_1}{\partial x^2} = \frac{p_1}{\tau_{p0}} \Rightarrow \frac{\partial^2 p_1}{\partial x^2} = \frac{p_1}{L_p^2}$$

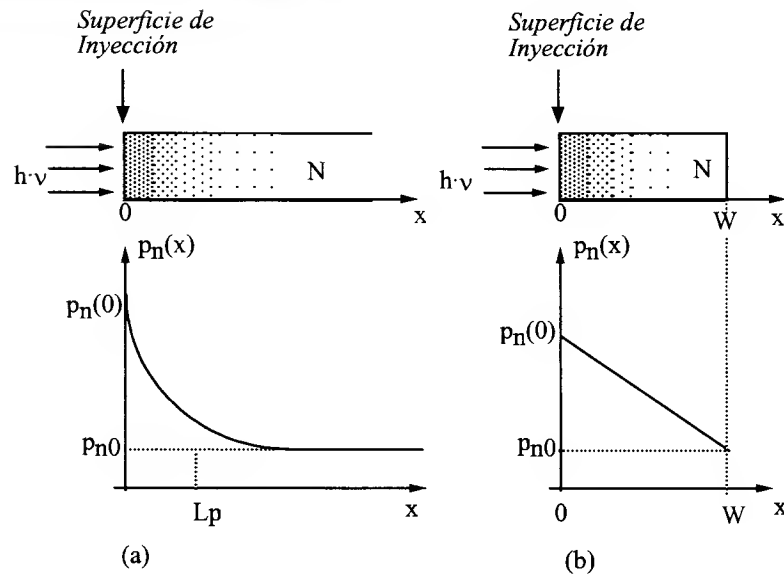
siendo  $L_p = \sqrt{D_p \cdot \tau_{p0}}$  la longitud de difusión.

La solución de esta ecuación es de la forma general

$$p_1 = p_n(x) - p_{n0} = A \cdot e^{x/L_p} + B \cdot e^{-x/L_p}$$

las condiciones de contorno son: 
$$\begin{cases} p_n(0) = cte \\ p_n(\infty) = p_{n0} \end{cases}$$

luego  $p_1(x) = p_n(x) - p_{n0} = [p_n(0) - p_{n0}] \cdot e^{-x/L_p} = p_1(0) \cdot e^{-x/L_p}$



**Figura A.8.** Solución particular de la ecuación de difusión para situaciones estacionarias en las que no interviene el tiempo. En  $x=0$  hay una inyección de valor constante que produce un exceso de minoritarios  $p_n(x)$  de forma exponencial. En el caso ideal **(a)** el exceso se atenúa al penetrar en el material, tendiendo a cero para  $x$  muy grande. En **(b)**, caso más próximo al real se interrumpe el proceso para  $x=W$  donde se extrae el exceso de carga.

Si cambiamos la segunda condición de contorno de forma que los portadores que lleguen a  $x = W$  se extraen, quedando de nuevo  $p_n(W) = p_{n0}$ , la solución de la ecuación anterior será de la forma:

$$p_n(x) = p_{n0} + [p_n(0) - p_{n0}] \frac{Sh \frac{W-x}{L_p}}{Sh \frac{W}{L_p}}$$

La existencia de un gradiente de concentración está siempre asociada a una corriente de difusión. Así por ejemplo la densidad de corriente en  $x = W$ , será:

$$J_p = -qD_p \left. \frac{\partial p}{\partial x} \right|_W$$

A lo largo del tema que precede a este apéndice hemos visto que este resultado es básico para el cálculo de las corrientes en un transistor bipolar. De hecho, este ejemplo corresponde al estudio de los minoritarios en la zona neutra de base. En la parte (a) de la *figura A.8* se ilustran los resultados correspondientes a la inyección de portadores en un extremo de una barra de material semiconductor de longitud infinita. En la parte (b) de la misma figura se muestran los resultados cuando la barra posee una longitud finita ( $W$ ).

### **A.3.5. Situaciones de no Equilibrio. Recombinación**

Tal como comentábamos en el apartado anterior en las situaciones de no equilibrio se rompe el balance dinámico y el sistema entra en un transitorio en el que las densidades de portadores son función del tiempo siendo además su producto superior o inferior al valor intrínseco dependiendo de que se trate de una situación de inyección o extracción de portadores.

Si el estímulo externo se mantiene constante el semiconductor alcanzará un nuevo estado estacionario con una nueva distribución de portadores resultado del nuevo balance entre generación y recombinación donde ahora la generación incluye el estímulo externo. Al eliminar el estímulo externo aparece un nuevo transitorio en el que las densidades son función del tiempo hasta que se restaura el equilibrio térmico. En este apartado final del apéndice estamos interesados en mencionar los procesos de recombinación asociados a esta vuelta al equilibrio. En un semiconductor se puede distinguir los siguientes procesos básicos de recombinación:

a) *En el volumen.*

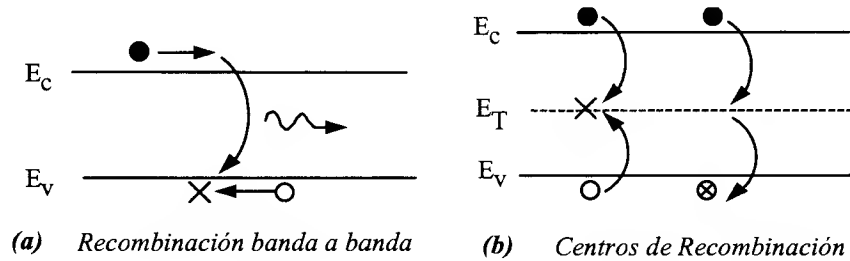
a.1. *Recombinación intrínseca (banda a banda).*

a.2. *Recombinación extrínseca (a través de un centro de recombinación con energía situada en el intervalo de energías prohibidas,  $E_g$ ).*

b) *En la superficie, (en las estructuras MIS, MOS y CCD).*

En los procesos de recombinación intrínseca (banda a banda), un electrón de la banda de conducción pasa directamente a un estado no ocupado de la banda de valencia, tal como se ilustra en la *figura A.9*. Este proceso es radiativo en el sentido de que el exceso de energía liberada durante la recombinación aparece en forma de luz con la emisión de un fotón. En la recombinación extrínseca intervienen niveles energéticos asociados a impurezas o defectos del cristal que actúan como centros intermedios de recombinación facilitando el salto a electrones y huecos. Por ejemplo, un electrón es

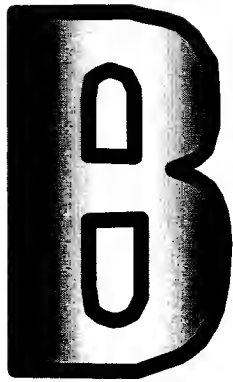
primero atrapado por un centro de impurezas o por una imperfección de la red y en una segunda etapa un hueco se recombina con este electrón. La recombinación intrínseca es una propiedad inherente al semiconductor, mientras que la extrínseca depende en gran manera de las imperfecciones de la red y es en general más probable.



**Figura A.9.** Visualización en la representación por bandas de los procesos de recombinación. (a) Intrínseca, banda a banda. (b) Extrínseca, a través de centros en el "gap".







---

## ***Apéndice B: Símbolos Lógicos***

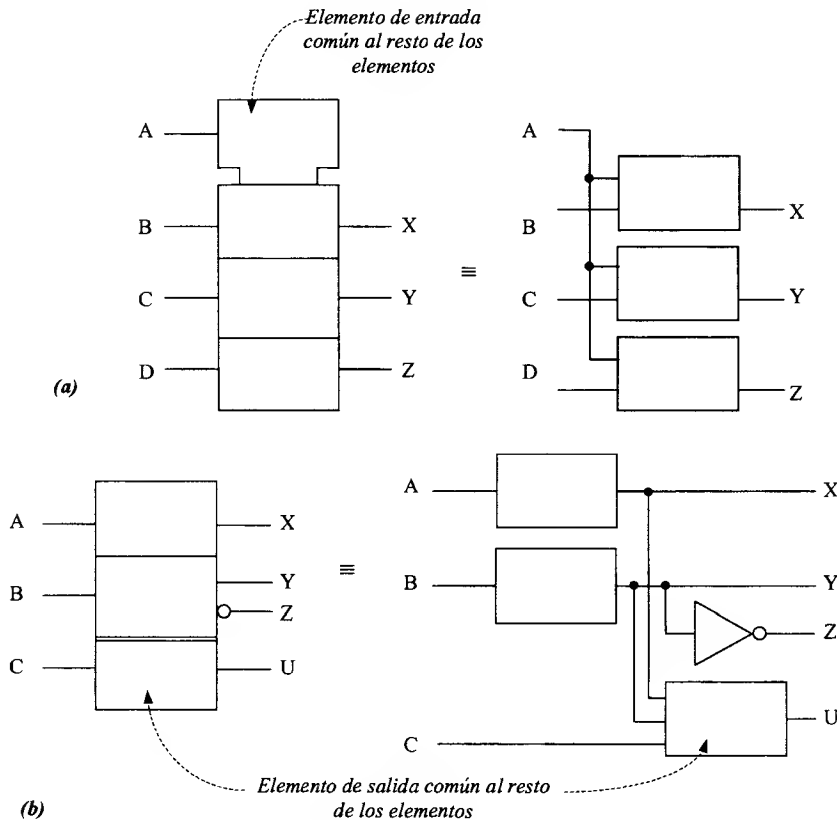
---

Incluimos en este apéndice un resumen del significado de los símbolos lógicos usados para describir externamente los distintos circuitos a los que se hace referencia en el texto.

El origen de este lenguaje de símbolos fue la propuesta inicial de la IEC (International Electrotechnical Commission) y el estándar del IEEE que buscaban una forma consensuada de describir las dependencias entre las entradas y las salidas de un circuito digital sin necesidad de mostrar de forma explícita el circuito interno.

Cuando un circuito posee una parte común a los distintos elementos que lo componen, esta parte se dibuja como bloque de control, tal como se ilustra en la *figura B.1.a.* y lo mismo ocurre con las salidas. Cuando hay un elemento común, se distingue gráficamente, tal como se muestra en la

figura B.1.b. En todas las otras circunstancias, el circuito se representa mediante un rectángulo, sea cual fuere su función.

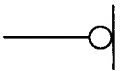
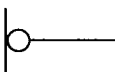
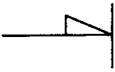
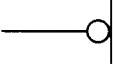
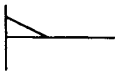
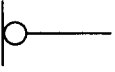
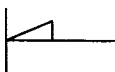
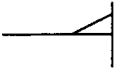

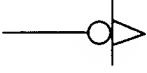


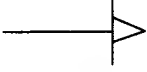




**Figura B.1.** (a) Representación del elemento común al resto de los elementos de un circuito. (b) Elemento de salida común al resto de los elementos del circuito. Ambos pueden incluir símbolos cualificadores que denoten su función lógica.

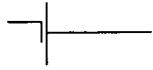
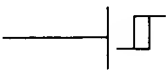


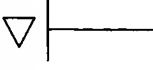

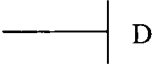
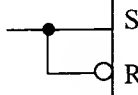
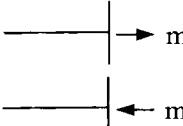
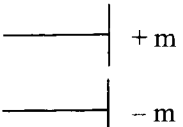
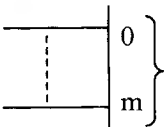
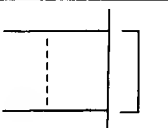
Los símbolos cualificadores usados en este libro han sido introducidos en los temas correspondientes y son, en general, de fácil comprensión (MUX= multiplexo, DMUX = demultiplexo, ALU, ROM, RAM, FIFO, etc...), por lo que no los repetimos aquí.

Sí que vamos a describir el significado de los símbolos de entrada y salida y algunos de los símbolos usados en el interior del rectángulo. Lo hacemos en forma de tablas. Los símbolos de

entrada y salida se muestran en la tabla de la *figura B.2*, y los correspondientes al interior del rectángulo en la tabla de la *figura B.3*.

	Negación lógica en la entrada. Un “0” externo produce un “1” interno.		
	Negación lógica en la salida. Un “1” interno produce un “0” externo		
	Entrada activa en baja. Equivalente en lógica positiva a		
	Salida activa en baja Equivalente en lógica positiva a		
	Entrada activa en baja, cuando el flujo de señales va de derecha a izquierda.		
	Salida activa en baja, cuando el flujo de señales va de derecha a izquierda		
	Señal bidireccional		
	Señales dinámicas activas en las transiciones que se indican	Lógica positiva 1 	Lógica negativa 1 
		0 	0 

**Figura B.2.** Símbolos de entrada y salida.

	Salida postpuesta de un biestable disparado por pulso. La salida cambia cuando la entrada que ha iniciado el cambio vuelve a su estado inicial (por ejemplo: Biestable J-K master-slave)	
	Entrada con histéresis. Es decir, con dos umbrales distintos, uno para cuando la señal va aumentando de valor y otro para cuando va disminuyendo	
	Salida en colector abierto para un transistor NPN, por ejemplo. Necesita apoyarse externamente a la tensión de alimentación usando una resistencia	
	Salida en emisor abierto. Necesita apoyarse externamente a la tierra usando una resistencia	
	Salida de tres estados	
	Entrada de facilitación (Enable input)	
	Dato de entrada a un circuito biestable. Equivalente a	
	Entradas de desplazamiento a la derecha (izquierda) $m=1, 2, 3$ , etc... Si $m=1$ , no suelen dibujarse.	
	Entradas de contar hacia arriba (hacia abajo) $m=1, 2, 3$ , etc... Si $m=1$ , no suelen dibujarse.	
	Agrupamiento binario ( $m=2^m$ )	
	Agrupamiento de líneas de entrada. Indica que dos o más terminales se usan para implementar una única línea lógica.	

**Figura B.3.** Símbolos interiores al rectángulo.

## RELACIONES DE DEPENDENCIA

El convenio de representación gráfica de los significados de las señales de entrada y salida a un circuito necesita complementarse ahora con otro conjunto de símbolos que representan las interrelaciones entre entradas, entre salidas y entre entradas y salidas. Estas relaciones se llaman “de dependencia”. En general estas relaciones son unidireccionales, de forma que enlazan dos nodos en los que el nodo origen “afecta a ...” y el nodo destino “es afectado por...”.

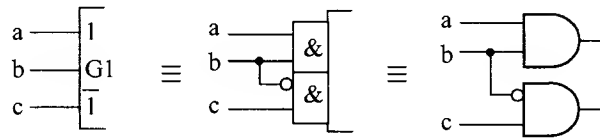
Se han considerado las siguientes relaciones de dependencia: **G**, para el AND, **V** para el OR, **N** para la *negación*, **Z** para la *interconexión*, **X** para la *transmisión*, **C** para el *control*, **R** y **S** para el *set* y *reset*, **EN** para *enable*, **M** para *modo* y **A** para *dirección* (address). La tabla de la *figura B.4* resume estas relaciones. Las letras usadas como símbolos (**A**, **C**, **G**, **M**, ...) aparecen junto a las entradas (o salidas) a las que afectan seguidas por un número. A su vez cada entrada (o salida) afectada por esa entrada se etiqueta con el mismo número. Esto no se aplica para **EN**, **R** y **S**.

<i>Tipo de Dependencia</i>	<i>Símbolo</i>	<i>Afectan a la Entrada en su estado “1”</i>	<i>Afectan a la Entrada en su estado “0”</i>
Address (dirección)	<b>A</b>	Permite la acción (dirección seleccionada)	Impide la acción (dirección no seleccionada)
Control	<b>C</b>	Permite la acción	Impide la acción
Enable (facilitación)	<b>EN</b>	Permite la acción	Impide la acción de las entradas Las salidas $\nabla$ en alta impedancia. No hay cambio en estado lógico interno.
AND	<b>G</b>	Permite la acción	Impone estado “0”
Modo	<b>M</b>	Permite la acción (modo seleccionado)	Impide la acción (modo no seleccionado)
Negate (X-OR)	<b>N</b>	Complementa el estado	No afecta
OR	<b>V</b>	Impone estado “1”	Permite la acción
Transmisión	<b>X</b>	Existe conexión bidireccional	No existe conexión bidireccional
Interconexión	<b>Z</b>	Impone estado “1”	Impone estado “0”

**Figura B.4.** Tabla resumen de la notación usada en las relaciones de interdependencia. No se incluyen *set* y *reset*.

## Relación de dependencia, G (AND)

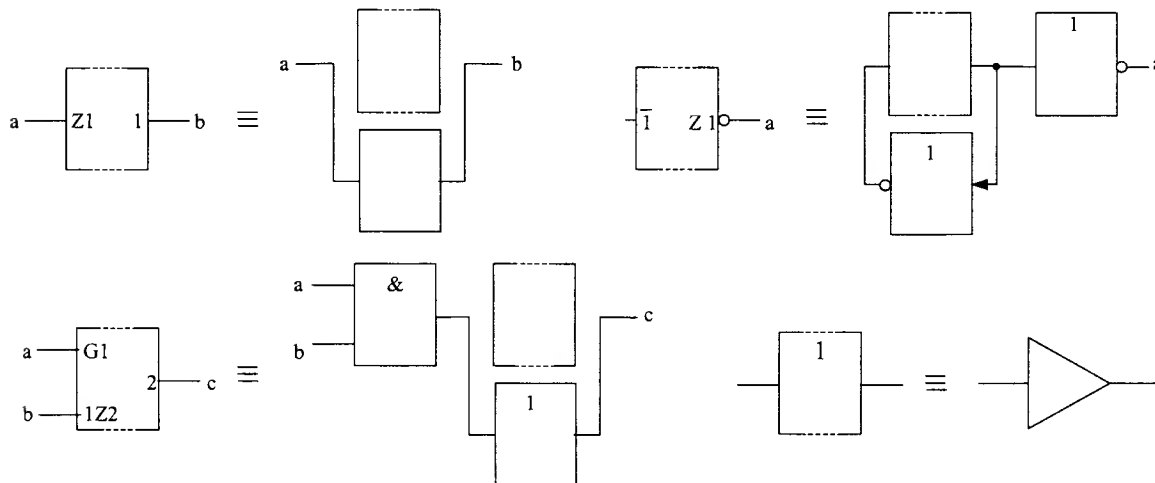
Las tres representaciones adjuntas son equivalentes. La letra **G** aparece en el interior del rectángulo seguida de un número (1 en el ejemplo) que se repite junto a las entradas a las que afecta.



La misma notación se usa para el caso OR, sustituyendo el símbolo **G** por el **V** tal como se describe en la tabla de la *figura B.4*.

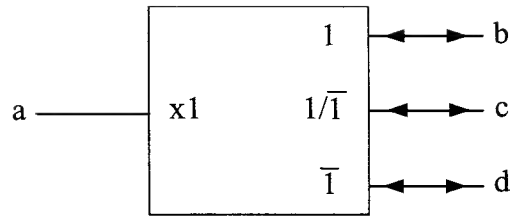
## Relación de interconexión, Z

Se usa para indicar la existencia de conexiones lógicas internas entre entradas, salidas, entradas internas y/o salidas internas. El estado lógico interno de una entrada o una salida afectadas por una entrada o salida **Zm** será el mismo que el de **Zm**, tal como se ilustra a continuación.



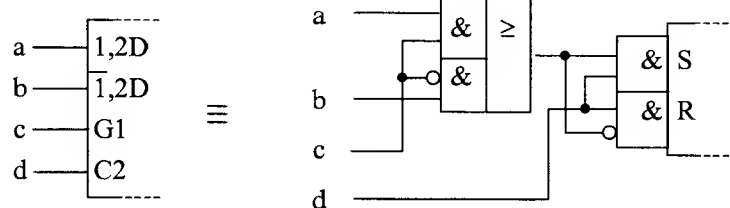
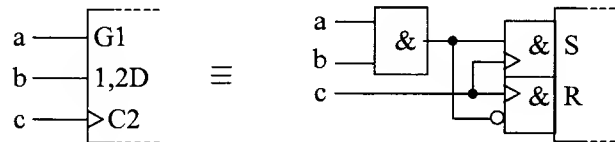
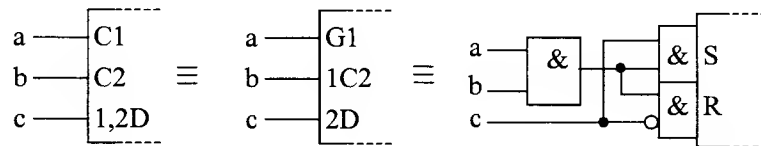
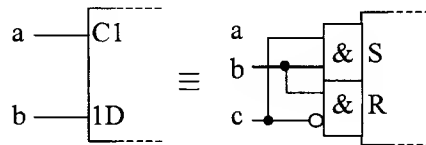
## Relación de transmisión, X

Se usa para representar conexiones bidireccionales controladas entre entradas y salidas. En el ejemplo adjunto, si **a=1** es que hay una conexión bidireccional efectiva entre **b** y **c**. Si **a=0**, es que hay una conexión bidireccional entre **c** y **d**.



## Relación de control, C

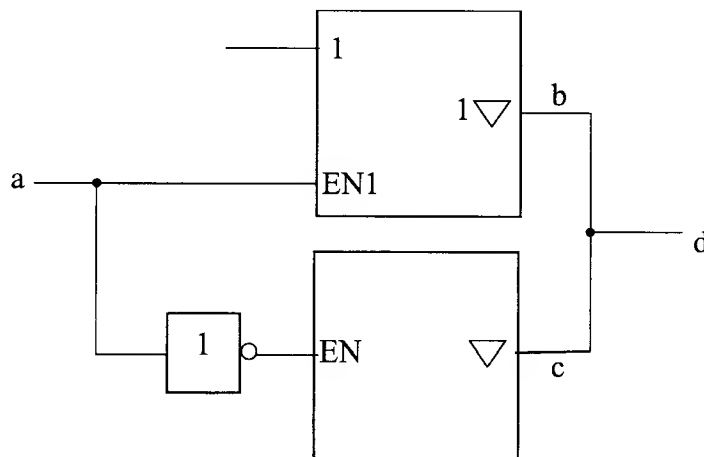
Las entradas de control se usan para facilitar o inhibir los datos de entrada de un circuito biestable (D, J-K, R.S), por ejemplo. Como pueden ser señales estáticas o dinámicas, esta circunstancia se tiene también en cuenta, tal como se ilustra en los ejemplos adjuntos. Cuando una entrada o salida **Cm** permanece en su estado interno "1" las salidas afectadas por esa **Cm** tienen el efecto normal sobre la función del elemento (por ejemplo, si es una señal de facilitación, **EN**, facilita las entradas). Cuando una entrada o salida **Cm** está a "0", las entradas afectadas por **Cm** quedan inhibidas y no afectan a la función del elemento.



## Relación de facilitación, EN

Una entrada **EN<sub>m</sub>** tiene el mismo efecto sobre las salidas que una entrada **EN** pero afecta sólo a aquellas salidas etiquetadas con el número de identificación **m**. También afecta a las entradas etiquetadas con el número **m**. En cambio, si en vez de **EN<sub>m</sub>** ponemos sólo **EN** esta entrada afecta a todas las salidas y a ninguna entrada.

La figura adjunta muestra un ejemplo de esta relación de dependencia. Cuando **a=0**, **b** está inhibida y **d=c**. En cambio, cuando **a=1**, **c** está inhibida y **d=b**.



En general, cuando **EN<sub>m</sub>=1**, las entradas afectadas tienen su efecto sobre la función del elemento tal como ha sido definida al especificar el circuito y las salidas afectadas por esta entrada permanecen en sus estados lógicos normales. Es decir, el circuito está facilitado.

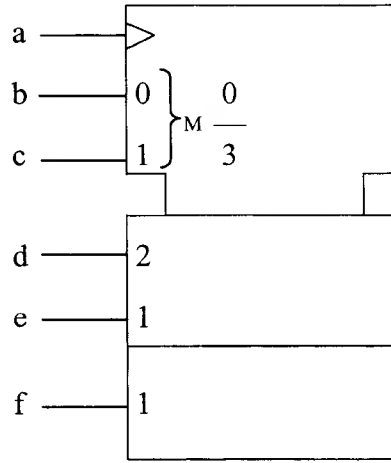
En cambio, cuando una entrada **EN<sub>m</sub>=0**, todas las entidades a las que afecta están inhibidas.

## Relación de modo, M

Cuando un circuito puede trabajar en distintos modos de funcionamiento, la relación de dependencia **M** se usa para indicar, en cada caso, cuáles son los efectos específicos de entradas y salidas sobre un elemento concreto del circuito. La figura adjunta muestra un ejemplo en el que la dependencia **M** afecta a las entradas. Hay dos entradas (**b** y **c**) que controlan los cuatro modos (0, 1, 2, 3). A su vez, las entradas **d**, **e** y **f** están sujetas al control dinámico de la entrada **a**, pero vamos a obviar este hecho para centrarnos en la descripción de la relación **M**. En *modo 0* (**b=0**, **c=0**), no hay



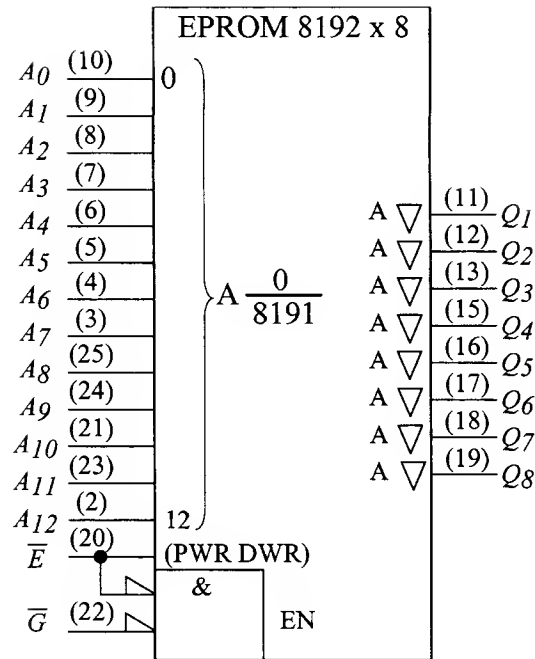
cambios. En *modo 1* ( $b=1$ ,  $c=0$ ) hay un proceso de carga paralelo a través de las entradas **e** y **f**. Por eso el número 1 aparece dentro del rectángulo al lado de **e** y **f**. En *modo 2* ( $b=0$ ,  $c=1$ ), hay, por ejemplo, desplazamiento y carga serie a través de la entrada **d**. Por eso aparece el número 2 al lado de **d**. Finalmente, en *modo 3* ( $b=c=1$ ) ocurre un proceso (contar, por ejemplo) que no afecta directamente a la facilitación o inhibición de las entradas **d**, **e** y **f**.



A lo largo del texto, cuando un circuito incluía la posibilidad de distintos modos de funcionamiento, lo hemos descrito en extenso usando tablas y sin hacer, en general, referencia a esta notación lógica.

### Relación de dependencia, A

El símbolo A se usa para denotar las dependencias de direccionamiento en circuitos de memoria, tal como los hemos descrito en el tema 7, 11 y 12. Por ejemplo, repetimos aquí el símbolo lógico de la EEPROM 8192x8 TMS-27C64 de la *figura 7.17* que numera externamente las líneas  $A_0$ , ...,  $A_{12}$  y engloba con un corchete en el interior del rectángulo las palabras direccionadas por esas líneas (de 0 a 8191). Las líneas de salida  $Q_0$ , ...,  $Q_8$  están afectadas por el direccionamiento, por lo que están etiquetadas con la letra A. Además, las salidas son de 3 estados, por lo que la A está seguida del símbolo  $\nabla$ . Obsérvese también el uso en este ejemplo de otros símbolos descritos previamente (entrada activa en baja y dependencia AND).



**Figura B.5.** Símbolo lógico de la memoria EPROM TMS27C64 usada para ilustrar la representación del direccionamiento.

## REFERENCIAS BIBLIOGRÁFICAS

1. Mann, F.A. *“Overview of IEEE Standard 91-1984. Explanation of Logic Symbols”*. Texas Instruments, 1984.
2. Sandige, R.S.: *“Modern Digital Design”*. McGraw-Hill. 1990.

